

응용 파일 시스템을 이용한 단일 실행 파일 이동형 SW 의 설계

최원혁, 김원영, 허성진
 한국전자통신연구원 SW 서비스연구팀
 e-mail : (whchoi, wykim, sjheo)@etri.re.kr

A Design of Portable Software that consist of single file using Application File System

Won-Hyuk Choi, Won-Young Kim, Sung-Jin Hur
 S/W Service Research Team, ETRI

요 약

본 논문은 임의의 호스트의 실행 환경과 리소스에 대한 변경 없이 독립적으로 응용 프로그램 수준의 가상화를 제공하는 단일 실행 파일을 가지는 이동형 SW 의 제공 방법에 대하여 소개한다. 이는 실행 포맷 변환기를 통하여 설치형 SW 에서 추출된 이동형 SW 정보를 본 논문에서 제공하는 응용 파일 시스템에 구축하고, 이를 실행하기 위하여 윈도우 기반 시스템에서 사용자 레벨의 라이브러리 후킹 기술을 이용한 가상 실행 라이브러리와, 이 가상 실행 라이브러리를 실제 구동되는 이동형 SW 프로세스에 바인딩하여 실행시키게 하는 실행 엔진 모두를 하나의 파일에 제공하는 방법 대하여 기술한다

1. 서론

인터넷의 대중화와 웹 어플리케이션을 이용한 서비스들의 보급 및 vmware[1]와 같은 가상화 솔루션들을 이용한 다양한 PC 사용환경의 출현으로 인하여 SW 의 유통방식에도 변화가 일어나고 있다. 이러한 변화의 일환으로 설치형 패키지 형태로 제공되는 SW 솔루션들이 온라인 서비스 형태로 제공되어 설치 없이 원 클릭만으로 사용 가능한 형태로 제공되고 있다.

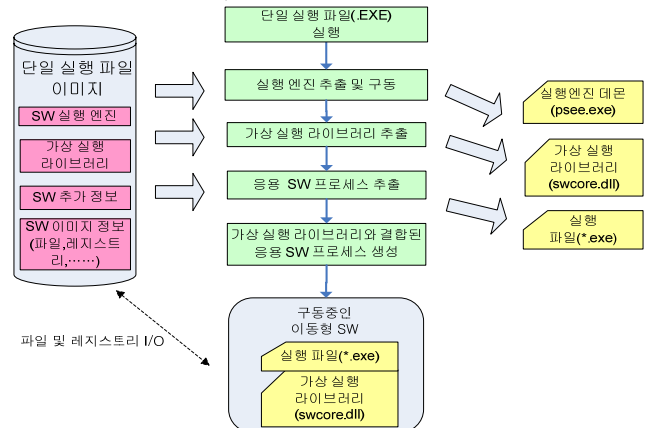
PortableApps[2]가 제공하는 이동형 SW 기술은 휴대용 메모리에 설치가 필요 없는 형태로 변환된 SW 를 저장하여 원 클릭으로 SW 를 실행 가능하게 한다. 소프트웨어는 스트리밍 방식으로 사용할 수 있는 온라인 솔루션인 Zpack[3] 외에, 사용자 PC 에 설치된 실행 엔진을 이용하여 변환된 SW 이미지를 원 클릭으로 실행하는 ipack[4]이라는 솔루션도 제공한다.

하지만, PortableApps 와 같은 솔루션은 패키지 SW 를 자동으로 변환하는 도구[5]가 제공되지 않고, 가공된 SW 도 여러 개의 파일로 나누어져 있다. 이는 보안 문제나 휴대 시에 불편함을 제공한다. 또한, ipack 솔루션은 독립적인 실행 엔진을 사용자의 PC 에 설치해야 하기 때문에 사용하려는 SW 의 설치 과정 대신에 자체 실행엔진을 설치 해야 하는 점이 불편하다.

이러한 기존 기술들이 가지는 보안성 및 휴대성은 자체 실행 엔진의 설치와 같은 단점을 해결하기 위하여 본 논문에서는 변환된 SW 이미지와 자체 실행 엔진을 제공하는 응용 파일 시스템에 구축하여 하나의 단일 파일로 묶어 배포 또는 실행함으로써 보안성 및 이동성 및 자체 실행엔진의 설치 과정을 없애고 원 클릭으로 SW 를 실행하는 방법에 대하여 기술한다.

2. 단일 실행 파일 이동형 SW

본 논문에서 제공하는 이동형 SW 는 아래의 (그림 1)의 왼쪽의 단일 실행파일 이미지로 구성된다. 단일 실행 파일 이동형 SW 는 윈도우에서 자체적으로 실행될 수 있는 EXE 파일과 같은 하나의 실행 파일로 구성된다. 구성은 SW 실행 엔진 때문, 사용자 수준의 후킹을 이용한 가상 실행 라이브러리, SW 추가 정보와 이동형 SW 의 파일 및 레지스트리와 같은 리소스를 자체적으로 제공하는 응용 파일 시스템에 구축한 SW 이미지 정보로 구성된다. 본 논문에서 제공하는 응용 파일 시스템은 가상화를 위하여 윈도우가 자체적으로 구성하는 파일 시스템을 이용하지 않고, 가상 실행 라이브러리와 연동하여 하나의 파일에서 동작할 수 있는 파일 시스템이다.



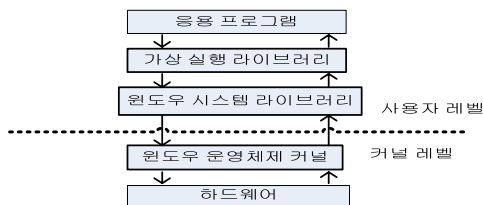
(그림 1) 단일 실행 파일 이동형 SW 동작 순서도

위의 (그림 1)은 단일 실행 파일을 실행시켰을 때, 단일 실행 파일이 실행되는 순서를 보여준다. 우선

이동형 SW 를 실행하면, SW 실행 엔진을 추출하여 이를 실행한다. 구동된 실행 엔진은 단일 실행 파일 이미지에서 가상 실행 라이브러리와 실제 실행되는 응용 SW 실행 파일을 추출하고 이를 바인딩하여 응용 SW 프로세스를 생성 및 구동시킨다. 구동된 이동형 SW 프로세스는 가상 실행 라이브러리를 통하여 단일 실행 파일 이미지의 응용 파일 시스템과 파일 및 레지스트리 I/O 를 수행하여 동작한다.

3. 가상 실행 라이브러리

가상 실행 라이브러리는 실행 엔진이 이동형 SW 의 실제 응용 프로그램 프로세스를 생성할 때, 직접 바인딩되어 실행 된다. 가상 실행 라이브러리는 실제 응용 프로그램이 실행될 때 발생하는 함수 호출을 사용자 수준에서 후킹하여 응용 파일 시스템에 구축된 SW 정보 이미지와 I/O 를 수행하게 한다. 아래 (그림 2)는 단일 실행 파일 이동형 SW 프로세스의 I/O 계층도이다. 일반적인 윈도우 응용 프로그램은 직접 윈도우 시스템 라이브러리의 호출을 통하여 동작하지만, 단일 실행 파일 이동형 SW 프로세스는 본 논문에서 제공되는 응용 파일 시스템을 이용한 가상화 기능을 제공하는 가상 실행 라이브러리를 통하여 윈도우 시스템 라이브러리 및 윈도우 운영체제 커널을 이용하여 하드웨어의 리소스와 I/O 를 수행한다.

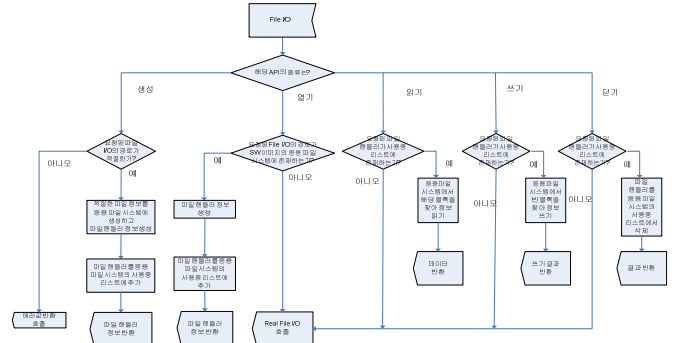


(그림 2) 이동형 SW 프로세스의 I/O 계층도

아래의 (그림 3)은 가상 실행 라이브러리에서 파일 I/O 처리를 위한 방법이다. 응용 프로그램이 CreateFile 과 같은 파일 I/O 함수를 호출하면, 매개 변수인 파일 경로가 적절한지를 판단하고, 파일 경로를 응용 파일 시스템에서 생성 시킬 수 있다면, 해당 파일 정보를 응용 파일 시스템에 만들고 관련된 파일 핸들러 정보를 생성한다. 생성된 파일 핸들러는 응용 파일 시스템의 사용중 리스트에 추가하고 생성된 파일 핸들러 정보를 해당 응용 프로그램으로 반환한다. 응용 프로그램이 OpenFile 과 같은 I/O 를 호출하면, 파일의 경로검사를 통하여 응용 파일 시스템에 존재하지 않는다면, 윈도우 시스템 라이브러리의 실제 파일 I/O 함수를 호출한다. 그렇지 않다면, 파일 핸들러 정보를 생성하고, 이를 응용 파일 시스템의 파일 정보와 연결시키고, 사용중 리스트에 추가하여 파일 핸들러 값을 반환한다. ReadFile 과 같은 함수가 호출되면, 매개변수로 전달되는 파일 핸들러가 응용 파일 시스템의 사용중 리스트에 존재하는가를 검사하고, 존재하면, 파일 핸들러에 연결된 응용 파일 시스템의 파일 정보를 이용하여 해당 블록을 찾아 읽고자 하는 데이터를 반환한다. 그렇지 않다면, 실제 파일 I/O 함

수를 호출한다. WriteFile I/O 도 ReadFile 과 마찬가지로 요청된 파일 핸들러가 사용중 리스트에 존재하는지를 검사하여, 파일 핸들러에 연결된 파일 정보를 이용하여 응용 파일 시스템의 적절한 블록을 찾아 정보를 쓰고, 결과를 반환한다. 파일을 사용하고 닫을 경우에도 파일 핸들러가 사용중 리스트에 존재하는지를 검사한다. 존재한다면, 파일 핸들러에 연결된 파일 정보를 제거하고, 사용중 리스트에서 삭제한다. 그렇지 않다면, 실제 파일 I/O 함수를 호출하여 처리한다.

마찬가지로 레지스트리 처리 관련 함수들도 응용 파일 시스템의 레지스트리 저장 영역을 이용하여 윈도우 레지스트리 하이브를 이용하지 않고 처리한다.



(그림 3) 가상 실행 라이브러리의 파일 I/O 처리도

4. 결론

본 논문에서는 응용 프로그램 수준의 가상화를 제공하는 단일 실행 파일로 구성된 이동형 SW 의 제공 방법에 대하여 소개하였다. 이는 설치형 SW 에서 추출된 SW 설치 정보를 본 논문에서 구현한 응용 파일 시스템에 구축하고, 이를 실행하기 위하여 가상 실행 라이브러리를 만들고, 이 가상 실행 라이브러리를 실제 구동되는 응용 SW 프로세스에 바인딩하여 실행시키게 하는 실행 엔진 데몬 모두를 하나의 파일에 제공하여 기존의 이동형 SW 솔루션들이 가지는 보안성 및 휴대성 또는 자체 실행 엔진의 설치와 같은 단점을 해결할 수 있다. 또한, 가상 실행 라이브러리를 커널 수준[6]에서 구현하지 않고, 사용자 수준에서 구현함으로써 이동형 SW 실행의 안정성을 제공한다.

참고문헌

[1] vmware, <http://www.vmware.com/>
 [2] PortableApps, <http://portableapps.com/>
 [3] Zpack, <http://www.z-pack.co.kr/>
 [4] ipack, <http://www.ipack4u.co.kr/>
 [5,6] 최원혁, 김원영, 최지훈, 최완, “가상화 지원 시스템과 Copy-on-Write 방법을 이용한 응용프로그램 가상화 방법의 설계 및 구현”, 『KOSTI 2007/한국콘텐츠학회 추계종합학술대회 논문집』 하권, 2007. pp807-811

본 연구는 지식경제부 및 정보통신연구진흥원의 IT 산업원천기술개발사업의 일환으로 수행하였음. [2007-S-015-01, SaaS 기반 이동형 개인맞춤 사무환경 구축 기술 개발]