

C# 기반의 윈도우 시작 프로그램 실시간 관리 솔루션 설계 및 구현

윤경섭*, 마성민⁰, 김동현*

⁰인하공업전문대학 컴퓨터정보과

e-mail : ksyoon@inhac.ac.kr*, jhaha199@naver.com⁰, bbkk999@naver.com*

C# based Windows Startup Program Real-Time Management Solution Design and Implementation

Kyung-Seob Yoon*, Sung-Min Ma⁰, Dong-Hyeon Kim*

⁰Dept. of Computer Science, Inha Technical College

● 요약 ●

본 논문에서는 윈도우 운영체제를 사용 간에 실시간으로 레지스트리에서 시작 프로그램을 감시하여 사용자 의도와 관계없이 실행되는 시작 프로그램의 등록을 방지하고 관리하고자 한다. 또한 데이터베이스와 연동하여 본 솔루션을 이용하는 사용자들에게 해당 시작 프로그램의 삭제 빈도를 백분율로 나타내어 시작 프로그램의 효율적인 관리에 도움을 줄 수 있도록 하였다. 이러한 기능을 강점으로 C# 기반의 시스템 환경 최적화 솔루션을 제시하고자 한다.

키워드: 레지스트리(Registry), 윈도우(Windows), 시작 프로그램(Startup), 스레드(Thread)

I. 서론

2015년 2월 KISA에서 보도한 국내 인터넷 이용환경 현황조사에 따르면 국내 컴퓨터 사용 인구의 97% 이상이 사용하는 운영체제가 마이크로소프트의 윈도우(Windows)이다.[1] 대중화된 윈도우 운영체제는 특정 프로그램들을 시작 프로그램으로 레지스트리에 등록하여 운영체제가 부팅됨과 동시에 실행되어진다. 이 프로그램들은 단순히 사용자가 이용하는 프로그램의 업데이트 등 단순 용도일 수도 있으나 애드웨어와 같은 악의적인 용도일 수도 있으며, 사용자들과 관계없이 실행됨으로써 시스템 성능 저하의 원인을 제공 할 수 있다.

이러한 문제를 방지하고자 시작 프로그램을 관리하는 최적의 솔루션을 설계 및 구현함으로써 시스템 리소스의 효율적인 관리가 가능하도록 한다.

II. 관련연구

일반 사용자들은 운영체제가 설치된 이후에 장기간 시작 프로그램을 관리하지 못하고 그 프로그램들을 누적시켜 윈도우 운영체제의 사용에 많은 불편함을 겪게 된다. 이를 해소하기 위한 시작 프로그램 관리 프로그램들을 표 1에서 비교해보았다.

표 1. 기존 시작 프로그램 관리 프로그램 비교

프로그램명	핵심기능
Custom StartUp	시작 프로그램의 추가 · 삭제 · 백업
Shock StartUp	시작 프로그램 활성화의 여부를 선택
Naver Cleaner	선택한 시작 프로그램을 삭제 시 재설치를 차단

기존 프로그램들을 비교한 결과 시작 프로그램에 대한 관리의 기능하나 실시간으로 추가된 새로운 시작 프로그램에 대한 관리는 불가능하였다. 따라서 기존 프로그램들의 기능에서 실시간으로 시작 프로그램을 탐지하고 관리하는 기능의 솔루션을 제공한다면 더 효율적인 시스템 관리가 가능할 것이다.

III. 본론

1. 시스템 구성

솔루션의 구현에 있어서 필요한 시스템 구성요소는 다음과 같다.

- 사용자 편의성을 위한 GUI 환경 구현
- 특정 레지스트리 값 변동에 따른 실시간 탐지
- 시작 프로그램의 삭제 현황 값이 저장되는 데이터베이스

실시간 감지를 최소한의 운영체제 리소스만으로 동작하기 위해서 최초 스캔 후 CPU 자원을 넘겨주고 일정시간 이후 다시 동작하는 형태로 구현되며, 변동 값이 탐지되면 사용자에게 탐지된 값을 표현하고 관리를 유도한다.

따라서 본 논문에서는 C#에서 지원하는 기능을 활용하여 솔루션을 구현하였다.

C#의 BackgroundWorker 컴포넌트를 이용하여 GUI와 비동기 방식으로 멀티쓰레드를 구성하였고, GUI를 가진 쓰레드와의 크로스 쓰레딩 문제 해결을 위하여 C#의 BeginInvoke라는 컨트롤들의 메소드를 사용하여 Backgroundworker 쓰레드와 UI 쓰레드의 연동을 비동기식으로 해결하였다.

또한 사용자가 기존에 가지고 있던 레지스트리 값과 일정시간 이후 변동 여부를 확인하기 위하여 배열 형태로 두 레지스트리 값을 C#의 IEnumerable 인터페이스를 이용하여 차집합을 구하였다.

2. 알고리즘

그림1, 그림2는 멀티쓰레드로 구현된 실시간 탐지에 대한 알고리즘을 나타낸 것이다.

그림1은 UI 쓰레드의 동작 알고리즘을 나타내고 탐지를 위한 BackgroundWorker 쓰레드를 실행시키며 추가된 시작프로그램에 대한 삭제 및 무시를 사용자가 선택하여 수행하도록 한다.

그림2는 BackgroundWorker 쓰레드의 알고리즘이다. 실시간으로 기존의 시작 프로그램 항목과 일정 시간이 지난 후의 항목들을 배열 형태로 저장하여 차집합 연산을 통해 비교한다. 이를 이용해서 얻은 결과를 UI 쓰레드에 제공하고 UI를 호출하도록 하여 사용자에게 제공되도록 한다.

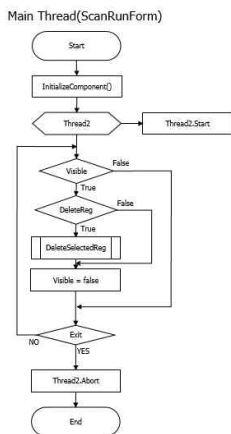


그림 1. 윈도우폼의 UI쓰레드 동작 알고리즘

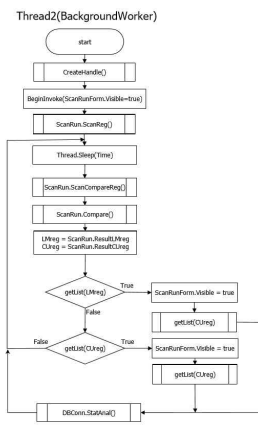


그림 2. BackgroundWorker쓰레드 동작 알고리즘

3. 구현

그림 3은 프로그램 초기 실행시의 메인 화면이다. 여기서는 시작 프로그램 등록에 대한 실시간 탐지 기능을 제외하고 솔루션에서 사용 가능한 기능들을 선택할 수 있다.



그림 3. 메인화면

그림 4는 시작 프로그램 관리 기능을 구현한 화면이다. 윈도우 운영체제에서는 현재 사용자 계정에 대한 시작 프로그램과 현재 사용자와 관계 없이 실행되는 시작 프로그램으로 구분한다. 구분된 항목을 사용자에게 제공하고 사용자는 이 화면에서 원하는 항목을 선택 후 삭제할 수 있다.



그림 4. 시작프로그램 관리 화면

그림 5는 메인화면이 최소화될 시 작업표시줄에 나타나는 트레이 아이콘의 구현 화면이다. 이 화면에서 실시간 감지를 작동시킬 수 있다.

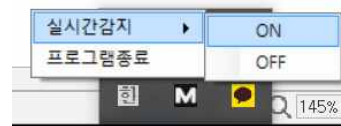


그림 5. 트레이 아이콘

그림 6은 새로 추가된 시작 프로그램이 탐지되었을 경우 사용자에게 알려주고 추가/삭제 판단을 요구한다. 또한 사용자들의 시작 프로그램 삭제 기록을 가지고 있는 데이터베이스에서 삭제 빈도를 불러와 계산하여 함께 나타낸다.

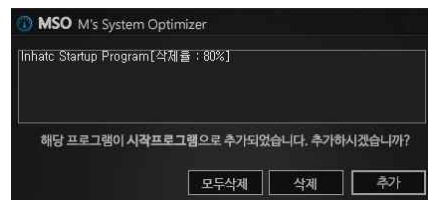


그림 6. 시작프로그램 실시간 탐지 화면

IV. 결론

본 논문에서는 다른 최적화 프로그램이 갖는 문제점인 실시간 감시가 불가능하다는 점을 해결하기 위한 시작 프로그램 실시간 관리 솔루션을 설계 및 구현하였다.

이 솔루션은 C#을 기반으로 설계 및 구현되었기 때문에 크로스 플랫폼의 문제점이나 차집합 연산 등을 지원하는 장점이 있었다. 또한 데이터베이스를 활용한 사용자 기반의 프로그램으로 다수의 사용자가 삭제한 값에 대하여 다른 사용자들에게 표현해줌으로써 잘 모르는 프로그램에 대한 추가 및 삭제 선택이 용이하다는 장점을 도출하였다.

결과적으로는 불필요한 시작 프로그램을 제거하고 실시간으로 사용자가 원하지 않는 시작 프로그램의 레지스트리 값 등록을 방지할 수 있어 평소 관리에 소홀한 일반 사용자도 쉽게 판단하여 삭제가 가능하다.

Reference

- [1] <http://www.kisa.or.kr>
- [2] <http://msdn.microsoft.com/>
- [3] <http://www.devpia.com/>
- [4] Sanghyun Park, "C# 4.0 Programming that stimulates the brain", Hanbit Media Inc, 2011