Journal of Next-generation Convergence Information Services Technology
차세대컨버전스정보서비스기술논문지
Vol.8, No.1, March (2019), pp.17-22
http://dx.doi.org/10.29056/jncist.2019.03.02

안드로이드 플랫폼기반의 응용 프로그램 제어 시스템 개발

Development of Application Control System based on Android Platform

배재환[1]

Jae-Hwan Bae[1]

요약

본 논문은 안드로이드 플랫폼 스마트폰에서 특정 응용 프로그램을 실행시켜 다른 응용 프로그램의 잠금을 해제하도록 하고 일정시간 경과 후 다시 잠금을 설정하는 장치 및 방법에 관한 것이다. 본 논문에 따른 스마트 응용 프로그램의 잠금/해제 방법은 스마트폰에 설치된 애플리케이션 프로그램은 잠금을 해제하고 다수의 다른 애플리케이션 프로그램은 잠금으로 설정하는 설정단계와, 애플리케이션 프로그램에 대한 실행명령이 입력되면 애플리케이션 프로그램을 실행하는 실행단계와, 애플리케이션 프로그램의 실행 중에 사용자로부터 입력되는 정보가 미리 설정된 애플리케이션 프로그램의 잠금 해제 조건을 만족하는지를 판단하는 판단단계와, 그리고 잠금해제 조건을 만족하면 상기 잠금으로 설정된 다수의 다른 애플리케이션 프로그램의 잠금을 해제하는 해제단계를 포함한다. 이를 통해서 스마트 응용프로그램 기술 발전에 도움이 될 것으로 기대한다.

핵심어 : 앱 잠금, 앱 해제, 안드로이드 플랫폼, 스마트 앱

Abstract

This paper describes an apparatus and method for executing a specific application program on an Android platform smartphone to release a lock of another application program and setting a lock after a predetermined time elapses. The method of locking / unlocking a smart application according to the present invention includes: a setting step of unlocking an application program installed in a smart phone and setting a plurality of other application programs as a lock; A determination step of determining whether information input from a user during execution of the application program satisfies an unlock condition of a predetermined application program; and a determination step of determining whether or not a plurality of other application programs And releasing the lock. We hope that this will help advance smart application technology.ol System based on Android Platform

Keyword : App lock, Un-app, Android platform, Smart app

1 Department of Game Engineering, TongMyong University, Busan, Korea [Professor]
e-mail: bjhmail@tu.ac.kr

Received(January 15, 2019), Review Result(1st: February 02, 2019), Accepted(March 08, 2019), Published(March 31, 2019)

ISSN: 2384-101X JNCIST
ⓒ 2019 NCISS

1. 서론

최근 수요가 폭발적으로 급증하고 있는 스마트폰(smart phone)은 휴대폰과 개인휴대단말기(personal digital assistance:PDA)의 장점을 통합시킨 것으로 휴대전화에 인터넷 통신과 정보검색 등 컴퓨터 지원기능을 추가한 지능형 단말기이다. 완제품으로 출시되어 주어진 기능만 사용하던 기존의 휴대폰과는 스마트폰은 방대한 종류의 다양한 애플리케이션(application) 프로그램을 사용자가 원하는 대로 설치하고 추가 또는 삭제할 수 있다는 점이 가장 큰 특징이다. 또한, 스마트폰은 무선인터넷을 이용하여 인터넷에 직접 접속할 수 있을 뿐 아니라 여러 가지 브라우징 프로그램을 이용하여 다양한 방법으로 접속할 수 있고 사용자가 원하는 애플리케이션을 직접 제작할 수도 있고 다양한 애플리케이션을 통하여 자신에게 알맞은 인터페이스를 구현할 수 있으며, 같은 운영체제(OS)를 가진 스마트폰간에 애플리케이션을 공유할 수 있는 점 등도 기존 휴대폰에 비해 큰 장점으로 꼽힌다. 2008년 구글의 안드로이드(Android)를 운영체제(OS)로 하는 스마트폰이 출시되었다. 안드로이드는 휴대전화기를 비롯하여 모바일 장치를 위한 운영체제와 미들웨어, 그리고 핵심 애플리케이션 프로그램을 포함하고 있는 휴대기기용 소프트웨어 스택이다. 이러한 안드로이드는 개발자들이 자바(java)언어로 애플리케이션을 작성할 수 있게 하였으며, 컴파일된 바이트코드를 구동할 수 있는 런타임 라이브러리를 제공한다. 또한, 안드로이드 SDK를 통하여 애플리케이션 프로그램을 개발하기 위해 필요한 도구 및 API(application program interface)를 제공한다[1][2]. 이에 본 논문에서는 스마트 안드로이드 플랫폼 앱 잠금 및 해제 기술을 제안 하고자 한다. 논문의 주요 구성은 1장 서론, 2장 안드로이드 플랫폼, 3장 앱 잠금 및 해제시스템, 4장 결론으로 구성 된다.

2. 안드로이드 플랫폼

2.1 안드로이드 개요

안드로이드(Android)는 리눅스 커널을 기반으로 구글에서 제작한 스마트폰과 같은 모바일 플랫폼의 운영체제와 미들웨어 및 중요 애플리케이션이 포함된 소프트웨어 집합이다. 구글은 안드로이드 OS에 대해 리눅스 커널의 제너럴 퍼블릭 라이선스(GPL)를 따르고 있으며, 새로운 운영체제의 버전 공개와 동시에 소스를 공개하고 있다. 2016년 현재 안드로이드는 세계에서 가장 대표적인 오픈소스 플랫폼이며, 세계 최다 사용자를 보유한 운영체제다. 안드로이드(86%)와 iOS 점유율(14%)을 소수점 아래에서 반올림하고 합치면 100%이다. 나머지 모바일 운영 체제는 의미 없는 점유율(0.1% 이하)만을 차지하고 있다. 스마트폰 OS 점유율 3위인 윈도우즈의 점유율조차 0.1%까지 떨어져서

ⓒ 2019 NCISS

결국 MS에서도 모바일 운영 체제 시장 포기를 선언하였고 모바일 OS 시장은 안드로이드와 iOS로 개편이 끝났다. 다른 모바일 OS 시장 점유율 자료에 의하면 2017년 11월 4일을 기준으로 안드로이드 66.74%, iOS 31.46%로 둘이 합쳐 98.20%이고, 윈도우즈는 0.73%이다[3].

2.2 안드로이드 역사 및 아키텍쳐

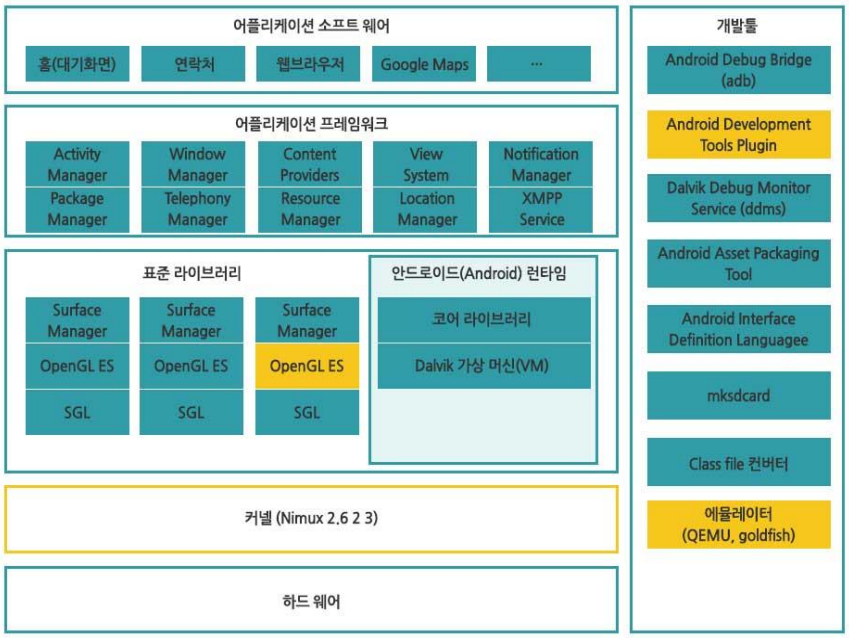
안드로이드(Android)가 휴대전화/태블릿 이외에도 갤럭시 기어와 같이 다양한 디바이스에 활용되었으나, 구글에서 웨어러블, 자동차 등에 맞춰 개량 버전을 출시했다. 2017년 2월 허니콤과 진저브레드의 구글 플레이 지원이 종료되었다. 코드네임은 A부터 시작하여 알파벳순으로 지어지며, 그 이름이 모두 디저트 이름이다[4]. 아래 [표 1]에 안드로이드 버전을 나타 내었다.

[표 1]안드로이드 버전
[Table 1] Android Version

코드네임	버전	릴리즈 날짜	API 버전	리눅스 커널
Oreo	8.0	2017년 8월 21일	API level 26	-
Nougat	7.1 - 7.1.2	2016년 10월 2일	API level 25	3.18
Nougat	7.0	2016년 8월 22일	API level 24	3.10
Marshmallow	6.0 - 6.0.1	2015년 10월 5일	API level 23	
Lollipop	5.1 - 5.1.1	2014년 11월 12일	API level 22	3.4.0
Lollipop	5.0 - 5.0.2		API level 21	
KitKat	4.4 - 4.4.4	2013년 10월 31일	API level 19	
Jelly Bean	4.3 - 4.3.1	2012년 7월 9일	API level 18	
Jelly Bean	4.2 - 4.2.2		API level 17	3.0.53
Jelly Bean	4.1 - 4.1.2		API level 16	3.0.31
Ice Cream Sandwich	4.0.3 - 4.0.4	2011년 10월 18일	API level 15, NDK 8	3.0.1
Ice Cream Sandwich	4.0 - 4.0.2		API level 14, NDK 7	
Honeycomb	3.2.X	2011년 2월 22일	API level 13	2.6.36
Honeycomb	3.1		API level 12, NDK 6	
Honeycomb	3.0		API level 11	
Gingerbread	2.3.3 - 2.3.7	2010년 12월 6일	API level 10	2.6.35
Gingerbread	2.3 - 2.3.2		API level 9, NDK 5	
Froyo(Frozen yogut)	2.2 - 2.2.3	2010년 5월 20일	API level 8, NDK 4	2.6.32
Eclair	2.1	2009년 10월 26일	API level 7, NDK 3	2.6.29
Eclair	2.0.1		API level 6	
Eclair	2.0		API level 5	
Donut	1.6	2009년 9월 15일	API level 4, NDK 2	
Cupcake	1.5	2009년 4월 27일	API level 3, NDK 1	2.6.27
Betit Four	1.1	2009년 2월 9일	API level 2	-

ⓒ 2019 NCISS

구글이 직접 개발에 참여하여 소프트웨어 지원 역시 전적으로 책임지는 제품은 구글 넥서스 시리즈가 있으며, 넥서스, 픽셀 시리즈 이외에도 구글이 직접적으로 개입하는 이른바 준 레퍼런스 기기들도 있다. 대표적으로 3.0 허니콤을 탑재한 갤럭시 탭 10.1과 Xoom이 있다. 하지만, 갤럭시 탭 10.1의 경우 삼성전자의 기기라는 느낌이 강한 반면, Xoom의 경우 사후지원 까지 구글이 전적으로 책임질 정도로 레퍼런스라는 느낌을 준다. 또한 2013년 들어서 이미 시장에 출시된 기기 중 구글 순정 안드로이드만 탑재하고 제조사 커스텀을 극한으로 최소화한 구글 플레이 에디션 모델들도 공개되었다. 현재까지 공개된 모델은 갤럭시 S4, ONE, 엑스페리아 Z 울트라가 있다. 다만, 제조사 커스텀이 어찌되었든 있기 때문에 구글이 서포트를 해주기는 하지만 운영체제 업그레이드는 각 제조사들이 책임진다고 한다. 안드로이드 L 부터는 안드로이드 원이라는 준 넥서스가 생긴다. 기존의 구글 플레이 에디션과의 차이점은 사양이 특정하게 지정되어 있으며 운영체제를 넥서스 마냥 구글에서 100% 관리한다는 것이다. 다른말로 설명하면 안드로이드 원은 윈도우폰처럼 되는 것이다. 지정된 사양내에서 제조사가 하드웨어를 자유롭게 만들고 운영체제는 구글이 책임진다. 한편, 넥서스 시리즈는 5x와 6p를 끝으로 단종 되었으며, 그 뒤를 픽셀 시리즈가 잇고 있다[5-9]. 안드로이드 플랫폼의 주요 아키텍쳐는 아래 [그림 1]과 같이 하드웨어, 커널, 표준 라이브러리, 안드로이드 런타임, 어플리케이션 프레임워크, 어플리케이션 소프트웨어 로 구성 되어 있다[5].



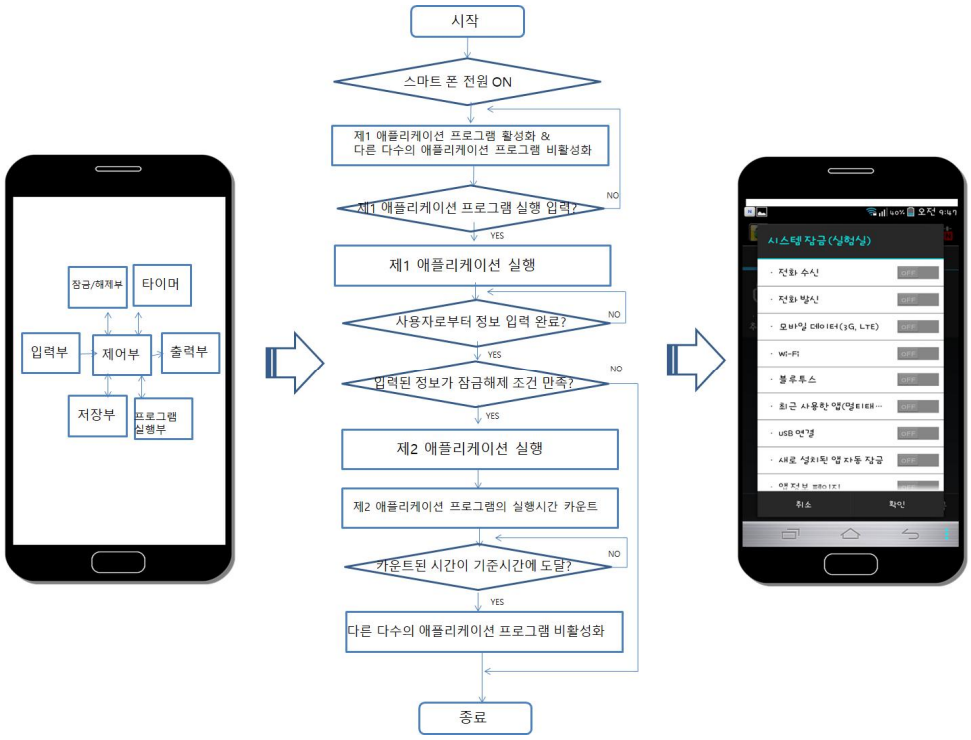
[그림 1] 안드로이드 구조

[Fig. 1] Android Architecture

ⓒ 2019 NCISS

3. 앱 잠금 및 해제시스템

안드로이드 플랫폼 스마트폰에서 특정 응용 프로그램을 실행시켜 다른 응용 프로그램의 잠금을 해제하도록 하고 일정시간 경과 후 다시 잠금을 설정하는 장치 및 방법에 관한 것이다. 스마트 응용 프로그램의 잠금/해제 방법은 스마트폰에 설치된 애플리케이션 프로그램은 잠금을 해제하고 다수의 다른 애플리케이션 프로그램은 잠금으로 설정하는 설정단계와, 애플리케이션 프로그램에 대한 실행명령이 입력되면 애플리케이션 프로그램을 실행하는 실행단계와, 애플리케이션 프로그램의 실행 중에 사용자로부터 입력되는 정보가 미리 설정된 애플리케이션 프로그램의 잠금 해제 조건을 만족하는지를 판단하는 판단단계와, 그리고 상기 잠금해제 조건을 만족하면 상기 잠금으로 설정된 다수의 다른 애플리케이션 프로그램의 잠금을 해제하는 해제단계를 포함한다. 아래 [그림 2]에 시스템 구성도 및 알고리즘을 나타내었다.



[그림 2]시스템 구성

[Fig. 2] System Configuration

주요 기대효과로는 스마트폰에서 다양한 애플리케이션 프로그램에 대하여 미리 잠금을 실행해 놓은 후 간단한 애플리케이션 프로그램의 실행을 통해 그 잠금을 해제하도록 함으로써 쉽고 간편

하게 스마트폰의 잠금을 해제할 수 있다. 또한, 게임에 무방비로 노출되어 있는 사용자에게 미리 설정된 시간 동안에만 게임을 할 수 있도록 함으로써 스마트폰에서의 게임시간을 조절할 수 있다. 그리고 스마트폰에서 일정한 시간이 경과하면 자동으로 애플리케이션 프로그램의 잠금을 실행할 수 있다.

4. 결론

본 논문은 안드로이드 플랫폼 스마트폰에서 특정 응용 프로그램을 실행시켜 다른 응용 프로그램의 잠금을 해제하도록 하고 일정시간 경과 후 다시 잠금을 설정하는 장치 및 방법에 관한 것이다. 본 논문에 따른 스마트 응용 프로그램의 잠금/해제 방법은 스마트폰에 설치된 애플리케이션 프로그램은 잠금을 해제하고 다수의 다른 애플리케이션 프로그램은 잠금으로 설정하는 설정단계와, 애플리케이션 프로그램에 대한 실행명령이 입력되면 애플리케이션 프로그램을 실행하는 실행단계와, 애플리케이션 프로그램의 실행 중에 사용자로부터 입력되는 정보가 미리 설정된 애플리케이션 프로그램의 잠금 해제 조건을 만족하는지를 판단하는 판단단계와, 그리고 잠금해제 조건을 만족하면 상기 잠금으로 설정된 다수의 다른 애플리케이션 프로그램의 잠금을 해제하는 해제단계를 포함한다. 이를 통해서 스마트 응용프로그램 기술 발전에 도움이 될 것으로 기대한다.

References

[1] Bae,Jae-Hwan, Apparatus and method for locking / unlocking application programs on a smartphone, Korean Patent Office Registration No. 10-2012-0019637, March 23 (2012).

[2] https://android-developers.googleblog.com/2008/09/announcing-android-10-sdk-release-1.html, Retrieved: March 11 (2018).

[3] https://www.androidcentral.com/google-sooner-prototype-appears-shows-one-googles-first-prototype-builds-android. Retrieved: November 9 (2018).

[4] https://tech-insider.org/mobile/research/2005/0817.html, Retrieved: March 12 (2018).

[5] https://www.pcworld.com/article/2034723/android-founder-we-aimed-to-make-a-camera-os.html, Retrieved: May 9 (2018).

[6] https://www.theverge.com/2013/4/16/4230468/android-originally-designed-for-cameras-before-smartphones, Retrieved: May 9 (2018).

[7] https://www.businessinsider.com/how-android-was-created-2015-3, Retrieved: May 9 (2018).

[8] https://www.engadget.com/2007/08/28/google-is-working-on-a-mobile-os-and-its-due-out-shortly/, Retrieved: March 11 (2018).

[9] https://www.theverge.com/2012/4/25/2974676/this-was-the-original-google-phone-presented-in-2006, Retrieved: March 12 (2018).

ⓒ 2019 NCISS