(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년02월26일
(11) 등록번호 10-1832594
(24) 등록일자 2018년02월20일

(51) 국제특허분류(Int. Cl.)
G06F 9/45 (2006.01) *G06F 11/14* (2006.01)
G06F 21/60 (2013.01) *G06F 9/44* (2018.01)
(52) CPC특허분류
G06F 8/40 (2013.01)
G06F 11/1402 (2013.01)
(21) 출원번호 10-2016-0019007
(22) 출원일자 2016년02월18일
심사청구일자 2016년02월18일
(65) 공개번호 10-2017-0097362
(43) 공개일자 2017년08월28일
(56) 선행기술조사문헌
KR1020130093775 A*
KR1020130027158 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
라인 가부시키가이샤
일본국 도쿄도 신주쿠쿠 신주쿠 4-1-6
(72) 발명자
정상민
경기도 성남시 분당구 황새울로360번길 42, 11층 (서현동,에이케이플라자분당점)
전상훈
경기도 성남시 분당구 황새울로360번길 42,11층(서현동, 에이케이플라자분당점)
(뒷면에 계속)
(74) 대리인
양성보

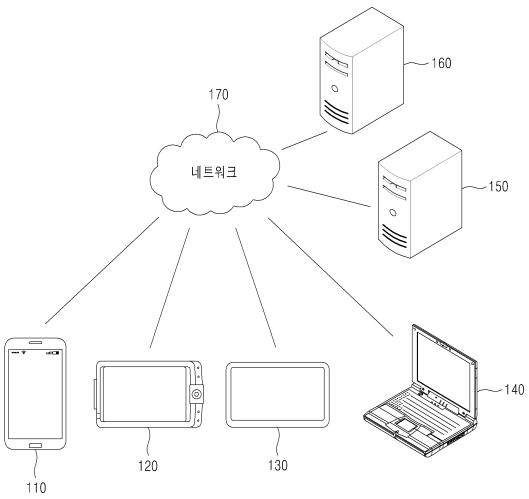
전체 청구항 수 : 총 16 항 심사관 : 서광훈

(54) 발명의 명칭 **중간 언어 파일의 로딩 속도 개선을 위한 방법 및 시스템**

(57) 요 약

중간 언어 파일의 로딩 속도 개선을 위한 방법 및 시스템이 개시된다. 컴퓨터에 의해 실행되는 로딩 속도 개선 방법은, 어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 단계, 상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계, 상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계 및 상기 중간 언어 파일의 변경 여부에 따라, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하거나 또는 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계를 포함할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06F 21/602 (2013.01)
G06F 8/443 (2013.01)
G06F 8/70 (2013.01)

(72) 발명자
정명주
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)
오왕진
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)
안성범
경기도 성남시 분당구 황새울로360번길 42,11층(서현동, 에이케이플라자분당점)
서동필
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)
한광희
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)
김태우
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)
임성열
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)
류주현
경기도 성남시 분당구 황새울로360번길 42, 11층(서현동,에이케이플라자분당점)

명 세 서

청구범위

청구항 1

컴퓨터로 구현되는 전자 기기와 결합되어 로딩 속도 개선 방법을 실행시키기 위해 기록매체에 저장된 컴퓨터 프로그램에 있어서,

상기 로딩 속도 개선 방법은,

어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 단계;

상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계;

상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계; 및

상기 중간 언어 파일이 변경된 경우 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하고, 상기 중간 언어 파일이 변경되지 않은 경우 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계

를 포함하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 2

제1항에 있어서,

상기 최적화 파일을 변경하여 저장하는 단계는,

상기 최적화 파일의 이름 및 경로 중 적어도 하나를 저장하고, 상기 최적화 파일의 이름 및 경로 중 적어도 하나의 정보를 변경하여 상기 최적화 파일을 변경하고,

상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계는,

상기 저장된 이름 및 경로 중 적어도 하나를 통해 상기 변경된 적어도 하나의 정보를 복원하여 상기 최적화 파일을 복원하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 3

제1항에 있어서,

상기 최적화 파일을 변경하여 저장하는 단계는,

상기 최적화 파일을 암호화하여 상기 최적화 파일을 변경하고,

상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계는,

상기 암호화된 최적화 파일을 복호화하여 상기 최적화 파일을 복원하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 4

제1항에 있어서,

상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계는,

상기 컴퓨터의 운영체제의 변경 여부를 확인하기 위한 정보 및 상기 어플리케이션의 변경 여부를 확인하기 위한 정보 중 적어도 하나의 정보를 더 생성하여 저장하고,

상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계는,

상기 저장된 적어도 하나의 정보를 더 이용하여 상기 운영체제의 변경 여부 또는 상기 어플리케이션의 변경 여부를 더 확인하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 5

제4항에 있어서,

상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계는,

상기 운영체제 및 상기 어플리케이션 중 적어도 하나가 변경된 경우에도, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 6

제1항에 있어서,

상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계는,

상기 중간 언어 파일에 대한 순환 중복 검사(Cyclic Redundancy Check)를 통해 오류검출코드를 생성 및 저장하고,

상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계는,

상기 다시 로딩된 중간 언어 파일에 대해 생성되는 오류검출코드와 상기 저장된 오류검출코드의 비교를 통해 상기 중간 언어 파일의 변경 여부를 확인하는 것을 특징으로 하는 컴퓨터 프로그램.

청구항 7

컴퓨터에 의해 실행되는 로딩 속도 개선 방법에 있어서,

어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 단계;

상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계;

상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계; 및

상기 중간 언어 파일이 변경된 경우 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하고, 상기 중간 언어 파일이 변경되지 않은 경우 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계

를 포함하는 것을 특징으로 하는 로딩 속도 개선 방법.

청구항 8

제7항에 있어서,

상기 최적화 파일을 변경하여 저장하는 단계는,

상기 최적화 파일의 이름 및 경로 중 적어도 하나를 저장하고, 상기 최적화 파일의 이름 및 경로 중 적어도 하나의 정보를 변경하여 상기 최적화 파일을 변경하고,

상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계는,

상기 저장된 이름 및 경로 중 적어도 하나를 통해 상기 변경된 적어도 하나의 정보를 복원하여 상기 최적화 파일을 복원하는 것을 특징으로 하는 로딩 속도 개선 방법.

청구항 9

제7항에 있어서,

상기 최적화 파일을 변경하여 저장하는 단계는,

상기 최적화 파일을 암호화하여 상기 최적화 파일을 변경하고,

상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계는,

상기 암호화된 최적화 파일을 복호화하여 상기 최적화 파일을 복원하는 것을 특징으로 하는 로딩 속도 개선 방

법.

청구항 10

제7항에 있어서,

상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계는,

상기 컴퓨터의 운영체제의 변경 여부를 확인하기 위한 정보 및 상기 어플리케이션의 변경 여부를 확인하기 위한 정보 중 적어도 하나의 정보를 더 생성하여 저장하고,

상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계는,

상기 저장된 적어도 하나의 정보를 더 이용하여 상기 운영체제의 변경 여부 또는 상기 어플리케이션의 변경 여부를 더 확인하는 것을 특징으로 하는 로딩 속도 개선 방법.

청구항 11

제10항에 있어서,

상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계는,

상기 운영체제 및 상기 어플리케이션 중 적어도 하나가 변경된 경우에도, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하는 것을 특징으로 하는 로딩 속도 개선 방법.

청구항 12

컴퓨터로 구현되는 전자 기기의 로딩 속도 개선 시스템에 있어서,

상기 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서

를 포함하고,

상기 적어도 하나의 프로세서는,

어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 최적화 파일 관리부;

상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 변경 확인 정보 관리부;

상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 변경 확인부; 및

상기 중간 언어 파일이 변경된 경우 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하고, 상기 중간 언어 파일이 변경되지 않은 경우 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 최적화 파일 재사용 제어부

를 포함하는 것을 특징으로 하는 로딩 속도 개선 시스템.

청구항 13

제12항에 있어서,

상기 최적화 파일 관리부는,

상기 최적화 파일의 이름 및 경로 중 적어도 하나를 저장하고, 상기 최적화 파일의 이름 및 경로 중 적어도 하나의 정보를 변경하여 상기 최적화 파일을 변경하고,

상기 최적화 파일 재사용 제어부는,

상기 저장된 이름 및 경로 중 적어도 하나를 통해 상기 변경된 적어도 하나의 정보를 복원하여 상기 최적화 파일을 복원하는 것을 특징으로 하는 로딩 속도 개선 시스템.

청구항 14

제12항에 있어서,

상기 최적화 파일 관리부는,

상기 최적화 파일을 암호화하여 상기 최적화 파일을 변경하고,

상기 최적화 파일 재사용 제어부는,

상기 암호화된 최적화 파일을 복호화하여 상기 최적화 파일을 복원하는 것을 특징으로 하는 로딩 속도 개선 시스템.

청구항 15

제12항에 있어서,

상기 변경 확인 정보 관리부는,

상기 컴퓨터의 운영체제의 변경 여부를 확인하기 위한 정보 및 상기 어플리케이션의 변경 여부를 확인하기 위한 정보 중 적어도 하나의 정보를 더 생성하여 저장하고,

상기 변경 확인부는,

상기 저장된 적어도 하나의 정보를 더 이용하여 상기 운영체제의 변경 여부 또는 상기 어플리케이션의 변경 여부를 더 확인하는 것을 특징으로 하는 로딩 속도 개선 시스템.

청구항 16

제15항에 있어서,

상기 최적화 파일 재사용 제어부는,

상기 운영체제 및 상기 어플리케이션 중 적어도 하나가 변경된 경우에도, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하는 것을 특징으로 하는 로딩 속도 개선 시스템.

발명의 설명

기 술 분 야

[0001] 아래의 설명은 중간 언어 파일의 로딩 속도 개선을 위한 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 중간 언어(Intermediate Language 또는 InterLanguage, IL)는 원시 언어 프로그램을 컴파일러로 번역하여 목적 언어 프로그램을 만들 때, 그 중간 단계로 거치게 되는 언어를 의미할 수 있다. 예를 들면 고수준 언어 프로그램을 어셈블리 언어로 바꾼 다음 그것을 어셈블하여 기계어 프로그램을 만든다면, 중간의 어셈블리 언어가 중간 언어가 될 수 있다.

[0003] 한국공개특허 제10-2007-0067953은 모바일 플랫폼의 중간 언어 변환 장치 및 그 방법에 관한 것으로, C 또는 C++ 언어로 개발된 모바일 플랫폼 소스 코드를 이동통신 단말기의 인터프리터에서 요구하는 중간 언어 코드로 변환하는 C/C++ 컴파일러와, 중간 언어 코드를 이동통신 단말기의 인터프리터에서 실행되는 포맷으로 변환하는 중간 언어 어셈블러에 대해 개시하고 있다.

[0004] 이러한 중간 언어로의 변환을 거치는 어플리케이션의 코드는 그 특성 상 디컴파일(decompile)에 취약하다. 예를 들어 자바(Java)와 같은 프로그래밍 언어로 제작된 어플리케이션의 코드는 중간 언어로의 변환을 거치는 특성 상 디컴파일에 취약하기 때문에 중요한 코드가 쉽게 노출되기도 하고, 코드 조작에도 매우 취약하다. 보다 구체적인 예로 자바로 작성된 코드는 일반적으로 클래스마다 확장자 ".class" 파일로 컴파일되고, 안드로이드(Android)에서는 자바로 작성된 코드들이 'classes.dex'와 같이 헤더와 데이터로 구성된 덱스(dex) 파일(자바 코드가 컴파일된 중간 언어 코드로서의 바이너리(binary) 코드인 바이트코드(bytecode))로 컴파일된다. 바이트코드는 자바 프로그램의 컴파일된 형태로서, 자바 프로그램이 바이트코드로 변환되면 네트워크를 통해 전송될 수 있고 가상머신(Virtual Machine, VM)에 의해 실행될 수 있다.

[0005] 예를 들어, 중간 언어 코드의 형태로 컴파일된 파일들이 서버로부터 전자 기기로 제공될 수 있고, 전자 기기는

지원하는 가상머신을 통해 중간 언어 코드의 형태로 컴파일된 파일들을 실행될 수 있다. 따라서 전자 기기에서는 중간 언어의 특성에 따라 중간 언어 코드의 형태로 컴파일된 파일들을 디컴파일하여 원래의 코드를 얻어서 수정하고, 수정된 코드를 중간 언어 코드의 형태로 다시 컴파일하는 것이 가능하기 때문에 어플리케이션을 위변조할 수 있다는 문제점이 존재한다.

[0006] 이러한 문제점을 해결하기 위해 종래기술에서는 덱스 파일과 같은 중간 언어 파일을 보호하고자 중간 언어 파일에 대한 난독화나 암호화 등의 기법을 적용한다.

[0007] 한편, 가상머신은 그 종류에 따라, 중간 언어 파일에 대한 최적화 파일을 생성하고, 최적화 파일을 로딩하여 처리하는 형태를 갖는다. 예를 들어, 안드로이드의 아트(ART) 모드의 가상머신은 어플리케이션이 설치될 때 어플리케이션에 포함된 중간 언어를 모두 번역하여 미리 최적화 파일(odex 파일 또는 oat 파일)을 생성하며, 최적화 파일을 이용하여 어플리케이션을 실행한다.

[0008] 그러나, 상술한 어플리케이션의 위변조 문제로 인해, 어플리케이션을 실행하는 시스템은 생성된 최적화 파일을 계속 이용할 수 없다는 문제점이 있다. 예를 들어, 시스템은 코드 보호 등의 이유로 최적화 파일을 삭제하고, 어플리케이션의 실행 시마다 중간 언어 파일에 대한 난독화나 암호화를 복호화하며, 어플리케이션의 매 실행 시마다 새로운 최적화 파일을 생성해야 한다. 이 경우, 최적화 파일의 생성 및 로딩에 따른 많은 시간이 소요된다는 문제점이 있다.

발명의 내용

해결하려는 과제

[0009] 중간 언어 파일(일례로, 안드로이드(Android)의 덱스(dex) 파일)에 대한 최적화 파일을 보호함과 동시에 재사용 가능하도록 함으로써 중간 언어 파일에 대한 로딩 속도를 개선할 수 있는 방법 및 시스템을 제공한다.

과제의 해결 수단

[0010] 컴퓨터로 구현되는 전자 기기와 결합되어 로딩 속도 개선 방법을 실행시키기 위해 기록매체에 저장된 컴퓨터 프로그램에 있어서, 상기 로딩 속도 개선 방법은, 어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 단계; 상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계; 상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계; 및 상기 중간 언어 파일의 변경 여부에 따라, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하거나 또는 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계를 포함하는 것을 특징으로 하는 컴퓨터 프로그램을 제공한다.

[0011] 컴퓨터에 의해 실행되는 로딩 속도 개선 방법에 있어서, 어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 단계; 상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 단계; 상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 단계; 및 상기 중간 언어 파일의 변경 여부에 따라, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하거나 또는 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 단계를 포함하는 것을 특징으로 하는 로딩 속도 개선 방법을 제공한다.

[0012] 컴퓨터로 구현되는 전자 기기의 로딩 속도 개선 시스템에 있어서, 상기 컴퓨터에서 판독 가능한 명령을 실행하도록 구현되는 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, 어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장하는 최적화 파일 관리부; 상기 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장하는 변경 확인 정보 관리부; 상기 중간 언어 파일이 다시 로딩되는 경우, 상기 저장된 정보를 이용하여 상기 중간 언어 파일의 변경 여부를 확인하는 변경 확인부; 및 상기 중간 언어 파일의 변경 여부에 따라, 상기 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하거나 또는 상기 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용하는 최적화 파일 재사용 제어부를 포함하는 것을 특징으로 하는 로딩 속도 개선 시스템을 제공한다.

발명의 효과

[0013] 중간 언어 파일에 대한 최적화 파일을 보호함과 동시에 재사용 가능하도록 함으로써 중간 언어 파일에 대한 로딩 속도를 개선할 수 있다.

도면의 간단한 설명

[0014] 도 1은 본 발명의 일실시예에 따른 네트워크 환경의 예를 도시한 도면이다.

도 2는 본 발명의 일실시예에 있어서, 전자 기기 및 서버의 내부 구성을 설명하기 위한 블록도이다.

도 3은 본 발명의 일실시예에 따른 전자 기기의 프로세서가 포함할 수 있는 구성요소의 예를 도시한 블록도이다.

도 4는 본 발명의 일실시예에 따른 전자 기기가 수행할 수 있는 방법의 예를 도시한 흐름도이다.

도 5는 본 발명의 일실시예에 있어서, 최적화 파일의 변경하는 예를 도시한 도면이다.

도 6은 본 발명의 일실시예에 있어서, 덱스 파일이 변경되지 않은 경우에 변경된 최적화 파일을 복원하는 예를 도시한 도면이다.

도 7은 본 발명의 일실시예에 있어서, 덱스 파일의 변경에 따라 새로운 최적화 파일을 생성하는 예를 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0015] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0016] 도 1은 본 발명의 일실시예에 따른 네트워크 환경의 예를 도시한 도면이다. 도 1의 네트워크 환경은 복수의 전자 기기들(110, 120, 130, 140), 복수의 서버들(150, 160) 및 네트워크(170)를 포함하는 예를 나타내고 있다. 이러한 도 1은 발명의 설명을 위한 일례로 전자 기기의 수나 서버의 수가 도 1과 같이 한정되는 것은 아니다.

[0017] 복수의 전자 기기들(110, 120, 130, 140)은 컴퓨터 장치로 구현되는 고정형 단말이거나 이동형 단말일 수 있다. 복수의 전자 기기들(110, 120, 130, 140)의 예를 들면, 스마트폰(smart phone), 휴대폰, 네비게이션, 컴퓨터, 노트북, 디지털방송용 단말, PDA(Personal Digital Assistants), PMP(Portable Multimedia Player), 태블릿 PC 등이 있다. 일례로 전자 기기 1(110)은 무선 또는 유선 통신 방식을 이용하여 네트워크(170)를 통해 다른 전자 기기들(120, 130, 140) 및/또는 서버(150, 160)와 통신할 수 있다.

[0018] 통신 방식은 제한되지 않으며, 네트워크(170)가 포함할 수 있는 통신망(일례로, 이동통신망, 유선 인터넷, 무선 인터넷, 방송망)을 활용하는 통신 방식뿐만 아니라 기기들간의 근거리 무선 통신 역시 포함될 수 있다. 예를 들어, 네트워크(170)는, PAN(personal area network), LAN(local area network), CAN(campus area network), MAN(metropolitan area network), WAN(wide area network), BBN(broadband network), 인터넷 등의 네트워크 중 하나 이상의 임의의 네트워크를 포함할 수 있다. 또한, 네트워크(170)는 버스 네트워크, 스타 네트워크, 링 네트워크, 메쉬 네트워크, 스타-버스 네트워크, 트리 또는 계층적(hierarchical) 네트워크 등을 포함하는 네트워크 토폴로지 중 임의의 하나 이상을 포함할 수 있으나, 이에 제한되지 않는다.

[0019] 서버(150, 160) 각각은 복수의 전자 기기들(110, 120, 130, 140)과 네트워크(170)를 통해 통신하여 명령, 코드, 파일, 컨텐츠, 서비스 등을 제공하는 컴퓨터 장치 또는 복수의 컴퓨터 장치들로 구현될 수 있다.

[0020] 일례로, 서버(150)는 전자 기기 2(120)로부터 네트워크(170)를 통해 업로드되는 어플리케이션의 패키지 파일을 등록하고, 등록된 패키지 파일에 보호 기능이나 암호화 기능 등과 같은 추가 기능을 위한 코드를 추가하여 패키지 파일에 대한 기능(일례로, 어플리케이션의 추가 기능)을 확장할 수 있다. 이때, 서버(150)는 직접 또는 별도의 서버(160)를 통해 전자 기기 1(110)로 기능이 확장된 패키지 파일을 제공할 수 있다. 전자 기기 1(110)은 어플리케이션의 패키지 파일을 통해 어플리케이션을 전자 기기 1(110)에 설치 및 구동할 수 있고, 어플리케이션을 통해 기설정된 서비스(일례로, 게임 서비스, 채팅 서비스, SNS(Social Network Service) 서비스 등)를 제공받을 수 있다.

[0022] 도 2는 본 발명의 일실시예에 있어서, 전자 기기 및 서버의 내부 구성을 설명하기 위한 블록도이다. 도 2에서는 하나의 전자 기기에 대한 예로서 전자 기기 1(110), 그리고 하나의 서버에 대한 예로서 서버(150)의 내부 구성을 설명한다. 다른 전자 기기들(120, 130, 140)이나 서버(160) 역시 동일한 또는 유사한 내부 구성을 가질 수 있다.

[0023] 전자 기기 1(110)과 서버(150)는 메모리(211, 221), 프로세서(212, 222), 통신 모듈(213, 223) 그리고 입출력 인터페이스(214, 224)를 포함할 수 있다. 메모리(211, 221)는 컴퓨터에서 판독 가능한 기록 매체로서,

RAM(random access memory), ROM(read only memory) 및 디스크 드라이브와 같은 비소멸성 대용량 기록장치(permanent mass storage device)를 포함할 수 있다. 또한, 메모리(211, 221)에는 운영체제와 적어도 하나의 프로그램 코드(일례로 전자 기기 1(110)에 설치되어 구동되는 브라우저나 영상 통화를 위한 어플리케이션 등을 위한 코드)가 저장될 수 있다. 이러한 소프트웨어 구성요소들은 드라이브 메커니즘(drive mechanism)을 이용하여 메모리(211, 221)와는 별도의 컴퓨터에서 판독 가능한 기록 매체로부터 로딩될 수 있다. 이러한 별도의 컴퓨터에서 판독 가능한 기록 매체는 플로피 드라이브, 디스크, 테이프, DVD/CD-ROM 드라이브, 메모리 카드 등의 컴퓨터에서 판독 가능한 기록 매체를 포함할 수 있다. 다른 실시예에서 소프트웨어 구성요소들은 컴퓨터에서 판독 가능한 기록 매체가 아닌 통신 모듈(213, 223)을 통해 메모리(211, 221)에 로딩될 수도 있다. 예를 들어, 적어도 하나의 프로그램은 개발자들 또는 어플리케이션의 설치 파일을 배포하는 파일 배포 시스템(일례로 상술한 서버(160))이 네트워크(170)를 통해 제공하는 파일들에 의해 설치되는 프로그램(일례로 상술한 어플리케이션)에 기반하여 메모리(211, 221)에 로딩될 수 있다.

[0024] 프로세서(212, 222)는 기본적인 산술, 로직 및 입출력 연산을 수행함으로써, 컴퓨터 프로그램의 명령을 처리하도록 구성될 수 있다. 명령은 메모리(211, 221) 또는 통신 모듈(213, 223)에 의해 프로세서(212, 222)로 제공될 수 있다. 예를 들어 프로세서(212, 222)는 메모리(211, 221)와 같은 기록 장치에 저장된 프로그램 코드에 따라 수신되는 명령을 실행하도록 구성될 수 있다.

[0025] 통신 모듈(213, 223)은 네트워크(170)를 통해 전자 기기 1(110)과 서버(150)가 서로 통신하기 위한 기능을 제공할 수 있으며, 다른 전자 기기(일례로 전자 기기 2(120)) 또는 다른 서버(일례로 서버(160))와 통신하기 위한 기능을 제공할 수 있다. 일례로, 전자 기기 1(110)의 프로세서(212)가 메모리(211)와 같은 기록 장치에 저장된 프로그램 코드에 따라 생성한 요청(일례로 영상 통화 서비스를 위한 요청)이 통신 모듈(213)의 제어에 따라 네트워크(170)를 통해 서버(150)로 전달될 수 있다. 역으로, 서버(150)의 프로세서(222)의 제어에 따라 제공되는 제어 신호나 명령, 컨텐츠, 파일 등이 통신 모듈(223)과 네트워크(170)를 거쳐 전자 기기 1(110)의 통신 모듈(213)을 통해 전자 기기 1(110)로 수신될 수 있다. 예를 들어 통신 모듈(213)을 통해 수신된 서버(150)의 제어 신호나 명령 등은 프로세서(212)나 메모리(211)로 전달될 수 있고, 컨텐츠나 파일 등은 전자 기기 1(110)가 더 포함할 수 있는 저장 매체로 저장될 수 있다.

[0026] 입출력 인터페이스(214, 224)는 입출력 장치(215)와의 인터페이스를 위한 수단일 수 있다. 예를 들어, 입력 장치는 키보드 또는 마우스 등의 장치를, 그리고 출력 장치는 어플리케이션의 통신 세션을 표시하기 위한 디스플레이와 같은 장치를 포함할 수 있다. 다른 예로 입출력 인터페이스(214)는 터치스크린과 같이 입력과 출력을 위한 기능이 하나로 통합된 장치와의 인터페이스를 위한 수단일 수도 있다. 보다 구체적인 예로, 전자 기기 1(110)의 프로세서(212)는 메모리(211)에 로딩된 컴퓨터 프로그램의 명령을 처리함에 있어서 서버(150)나 전자 기기 2(120)가 제공하는 데이터를 이용하여 구성되는 서비스 화면이나 컨텐츠가 입출력 인터페이스(214)를 통해 디스플레이에 표시될 수 있다.

[0027] 또한, 다른 실시예들에서 전자 기기 1(110) 및 서버(150)는 도 2의 구성요소들보다 더 많은 구성요소들을 포함할 수도 있다. 그러나, 대부분의 종래기술적 구성요소들을 명확하게 도시할 필요성은 없다. 예를 들어, 전자 기기 1(110)은 상술한 입출력 장치(215) 중 적어도 일부를 포함하도록 구현되거나 또는 트랜시버(transceiver), GPS(Global Positioning System) 모듈, 카메라, 각종 센서, 데이터베이스 등과 같은 다른 구성요소들을 더 포함할 수도 있다. 보다 구체적인 예로, 전자 기기 1(110)이 스마트폰인 경우, 일반적으로 스마트폰이 포함하고 있는 가속도 센서나 자이로 센서, 카메라, 각종 물리적인 버튼, 터치패널을 이용한 버튼, 입출력 포트, 진동을 위한 진동기 등의 다양한 구성요소들이 전자 기기 1(110)에 더 포함되도록 구현될 수 있음을 알 수 있다.

[0029] 본 발명의 실시예들에 따른 로딩 속도 개선 시스템은 어플리케이션이 실행되는 장치에 구현될 수 있다. 다시 말해, 본 발명의 실시예들에 따른 로딩 속도 개선 방법은 어플리케이션이 설치 및 구동되는 모든 장치에서 실행될 수 있다. 이후에서는 설명의 편의를 위해 전자 기기 1(110)에 어플리케이션의 설치 및 구동을 위한 패키지 파일이 저장되고, 전자 기기 1(110)에 구현되는 로딩 속도 개선 시스템을 통해 로딩 속도 개선 방법이 수행되는 실시예를 설명한다.

[0030] 도 3은 본 발명의 일실시예에 따른 전자 기기의 프로세서가 포함할 수 있는 구성요소의 예를 도시한 블록도이고, 도 4는 본 발명의 일실시예에 따른 전자 기기가 수행할 수 있는 방법의 예를 도시한 흐름도이다.

[0031] 전자 기기 1(110)은 로딩 속도 개선 시스템을 구현할 수 있으며, 도 3에 도시된 바와 같이 전자 기기 1(110)에 포함된 프로세서(212)는 구성요소들로서 로딩 제어부(310), 최적화 파일 관리부(320), 변경 확인 정보 관리부(330), 변경 확인부(340) 및 최적화 파일 재사용 제어부(350)를 포함할 수 있다. 이러한 프로세서(212) 및 프

로세서(212)의 구성요소들은 도 4의 로딩 속도 개선 방법이 포함하는 단계들(410 내지 450)을 수행하도록 전자 기기 1(110)을 제어할 수 있다. 이때, 프로세서(212) 및 프로세서(212)의 구성요소들은 메모리(211)가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다. 여기서, 프로세서(212)의 구성요소들은 전자 기기 1(110)에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서(212)에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 예를 들어, 프로세서(212)가 상술한 제어 명령에 따라 로딩 속도 개선 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리(221)에 로딩하도록 동작하는 기능적 표현으로 로딩 제어부(310)가 사용될 수 있다.

[0032] 단계(410)에서 로딩 제어부(310)는 로딩 속도 개선 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리(211)에 로딩할 수 있다. 예를 들어, 전자 기기 1(110)에서 프로그램이 실행되면, 로딩 제어부(310)는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리(211)에 로딩하도록 서버(150)를 제어할 수 있다.

[0033] 이때, 메모리(211)에 로딩되는 프로그램 코드는 어플리케이션을 위한 프로그램 코드와 최적화 파일을 관리하기 위한 제어 기능을 제공할 수 있다. 어플리케이션을 위한 중간 언어 파일 역시 단계(410)에서 로딩될 수 있다.

[0034] 프로세서(212)가 포함하는 최적화 파일 관리부(320), 변경 확인 정보 관리부(330), 변경 확인부(340) 및 최적화 파일 재사용 제어부(350) 각각은 메모리(211)에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(420 내지 450)을 실행하기 위한 프로세서(212)의 기능적 표현들일 수 있다.

[0035] 단계(420)에서 최적화 파일 관리부(320)는 어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장할 수 있다. 이미 설명한 바와 같이 중간 언어 파일이 로딩되는 경우, 중간 언어 파일을 번역한 최적화 파일이 생성되어 활용될 수 있다. 이때, 최적화 파일 관리부(320)는 생성된 최적화 파일을 변경하여 저장할 수 있다. 최적화 파일의 변경은 최적화 파일의 이름이나 경로를 변경하거나 또는 최적화 파일을 암호화하는 등과 같이 다시 원래의 최적화 파일을 획득할 수 있는 모든 방법들 중 적어도 하나를 대상으로 할 수 있다.

[0036] 예를 들어, 최적화 파일 관리부(320)는 생성된 최적화 파일의 이름 및 경로 중 적어도 하나를 저장하고, 최적화 파일의 이름 및 경로 중 적어도 하나를 변경하여 최적화 파일을 변경할 수 있다. 이후, 저장된 이름 및 경로 중 적어도 하나는 이후 변경된 최적화 파일을 복원하는데 활용될 수 있다.

[0037] 다른 예로, 최적화 파일 관리부(320)는 생성된 최적화 파일을 암호화하여 최적화 파일을 변경할 수 있다. 이후, 암호화된 최적화 파일은 복호화를 통해 복원될 수 있다. 암호화 방법 역시 다시 원래의 최적화 파일을 복원할 수 있는 모든 방법들 중 하나가 활용될 수 있다. 일례로, 암호화에 사용되는 키를 고정적으로 사용할 수도 있고, 개별 암호화 및 복호화의 인스턴스마다 새로운 키를 이용하여 암호화 및 복호화를 처리할 수도 있다. 키는 대칭키나 비대칭키 등, 그 종류와 무관하게 필요에 따라 선택된 종류의 키가 활용될 수 있다.

[0038] 단계(430)에서 변경 확인 정보 관리부(330)는 중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장할 수 있다. 일례로, 중간 언어 파일에 대한 순환 중복 검사(Cyclic Redundancy Check)를 통해 생성되는 오류검출코드가 중간 언어 파일의 변경 여부를 확인하기 위한 정보로서 생성될 수 있다. 중간 언어 파일의 변경 여부를 확인할 수 있는 방법이라면 제한되지 않고 활용될 수 있다.

[0039] 또한, 필요에 따라 선택적으로 운영체제(전자 기기 1(110)의 운영체제)의 변경 여부나 어플리케이션의 변경 여부를 확인하기 위한 정보가 더 생성 및 저장될 수 있다.

[0040] 단계(440)에서 변경 확인부(340)는 중간 언어 파일이 다시 로딩되는 경우, 저장된 정보를 이용하여 중간 언어 파일의 변경 여부를 확인할 수 있다. 상술한 순환 중복 검사의 예에서 변경 확인부(340)는 다시 로딩된 중간 언어 파일에 대해 생성되는 오류검출코드와 저장된 오류검출코드의 비교를 통해 중간 언어 파일의 변경 여부를 확인할 수 있다.

[0041] 또한, 변경 확인부(340)는 단계(440)에서 운영체제의 변경 여부나 어플리케이션의 변경 여부를 더 확인할 수도 있다. 이 경우 단계(430)에서 더 저장될 수 있는 정보(운영체제의 변경 여부나 어플리케이션의 변경 여부를 확인하기 위한 정보)가 운영체제의 변경 여부나 어플리케이션의 변경 여부를 확인하기 위해 활용될 수 있다.

[0042] 단계(450)에서 최적화 파일 재사용 제어부(350)는 중간 언어 파일의 변경 여부에 따라, 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하거나 또는 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용할 수 있다. 예를 들어 최적화 파일 재사용 제어부(350)는 중간 언어 파일이 변경되지 않는 동안에는 변경된

최적화 파일을 복원하여 재사용할 수 있다. 따라서 최적화 파일에 대한 중복 생성을 최소화할 수 있어 로딩 속도를 개선할 수 있다.

[0043] 중간 언어 파일이 변경되는 경우(일례로, 어플리케이션의 패치)에는 이전에 생성된 최적화 파일을 그대로 사용할 수 없기 때문에 변경되어 저장된 최적화 파일은 삭제하고, 다시 로딩된 중간 언어 파일을 통해 새로운 최적화 파일을 생성하여 사용할 수 있다. 이때, 다시 생성된 새로운 최적화 파일에 대해 단계(420) 내지 단계(450)이 반복적으로 수행됨으로써, 새로운 최적화 파일 역시 이후의 중간 언어 파일의 로딩 시에 재사용될 수 있다.

[0044] 또한, 최적화 파일 재사용 제어부(350)는 운영체제나 어플리케이션이 변경되는 경우에도 변경되어 저장된 최적화 파일을 삭제하고 새로운 최적화 파일을 생성할 수 있다. 다시 말해, 최적화 파일 재사용 제어부(350)는 중간 언어 파일, 운영체제 및 어플리케이션 중 적어도 하나가 변경된 경우, 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성할 수 있다. 이 경우, 최적화 파일 재사용 제어부(350)는 중간 언어 파일, 운영체제 및 어플리케이션이 변경되지 않는 동안은 변경된 최적화 파일을 복원하여 재사용할 수 있기 때문에 최적화 파일에 대한 중복 생성을 최소화할 수 있어 로딩 속도를 개선할 수 있다.

[0046] 도 5는 본 발명의 일실시예에 있어서, 최적화 파일의 변경하는 예를 도시한 도면이다. 전자 기기 1(110)에서 어플리케이션이 설치되거나 또는 구동되는 경우 어플리케이션의 안드로이드 응용 프로그램 패키지(Android application package, APK) 파일이 포함하는 덱스 파일(510)이 로더(520)에 의해 로딩될 수 있으며, 덱스 파일(510)에 대한 최적화 파일(530)이 생성될 수 있다.

[0047] 이때, 전자 기기 1(110)에 구현된 로딩 속도 개선 시스템(540)은 생성된 최적화 파일을 변경하여 저장할 수 있다. 도 5에서는 로딩 속도 개선 시스템(540)이 최적화 파일(530)을 암호화하여 변경하고, 암호화된 최적화 파일(550)을 저장하는 예를 나타내고 있다.

[0048] 이 경우, 로딩 속도 개선 시스템(540)은 변경 확인 정보(560)를 생성하여 저장할 수 있다. 변경 확인 정보(560)는 적어도 덱스 파일(510)의 변경 여부를 확인할 수 있는 정보를 포함할 수 있으며, 선택적으로 안드로이드 운영체제의 변경 여부를 확인하기 위한 정보 및/또는 어플리케이션의 변경 여부를 확인하기 위한 정보를 더 포함할 수 있다.

[0049] 도 6은 본 발명의 일실시예에 있어서, 덱스 파일이 변경되지 않은 경우에 변경된 최적화 파일을 복원하는 예를 도시한 도면이다. 도 6은 변경되지 않은 덱스 파일(510)이 로더(520)를 통해 다시 로딩되는 경우를 나타내고 있다. 로딩 속도 개선 시스템(540)은 다시 로딩되는 덱스 파일(510)의 변경 확인 정보를 생성하여 저장된 변경 확인 정보(560)와 비교함으로써 덱스 파일(510)의 변경 여부를 확인할 수 있다. 도 6의 예시에서는 덱스 파일(510)이 변경되지 않았기 때문에, 로딩 속도 개선 시스템(540)은 암호화된 최적화 파일(550)을 복호화하여 원래의 최적화 파일(도 5의 최적화 파일(530))을 복원할 수 있고, 복원된 최적화 파일을 재사용할 수 있다.

[0050] 도 7은 본 발명의 일실시예에 있어서, 덱스 파일의 변경에 따라 새로운 최적화 파일을 생성하는 예를 도시한 도면이다. 도 7은 변경된 덱스 파일(710)이 로더(520)를 통해 로딩되는 경우를 나타내고 있다. 로딩 속도 개선 시스템(540)은 로딩되는 변경된 덱스 파일(710)의 변경 확인 정보를 생성하여 저장된 변경 확인 정보(560)와 비교함으로써 덱스 파일(510)의 변경 여부를 확인할 수 있다. 도 7의 예시에서는 변경된 덱스 파일(710)이 로딩되었기 때문에, 로딩 속도 개선 시스템(540)은 암호화된 최적화 파일(550)을 삭제하고, 새로운 최적화 파일(720)을 생성할 수 있다.

[0051] 이후, 새로운 최적화 파일(720) 역시 변경(암호화)되어 저장될 수 있고, 변경된 덱스 파일(710)에 대한 변경 확인 정보 역시 새로 저장될 수 있다.

[0053] 이처럼 본 발명의 실시예들에 따르면, 중간 언어 파일에 대한 최적화 파일을 보호함과 동시에 재사용 가능하도록 함으로써 중간 언어 파일에 대한 로딩 속도를 개선할 수 있다.

[0055] 이상에서 설명된 시스템 또는 장치는 하드웨어 구성요소, 소프트웨어 구성요소 또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 어플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여,

데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0056] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

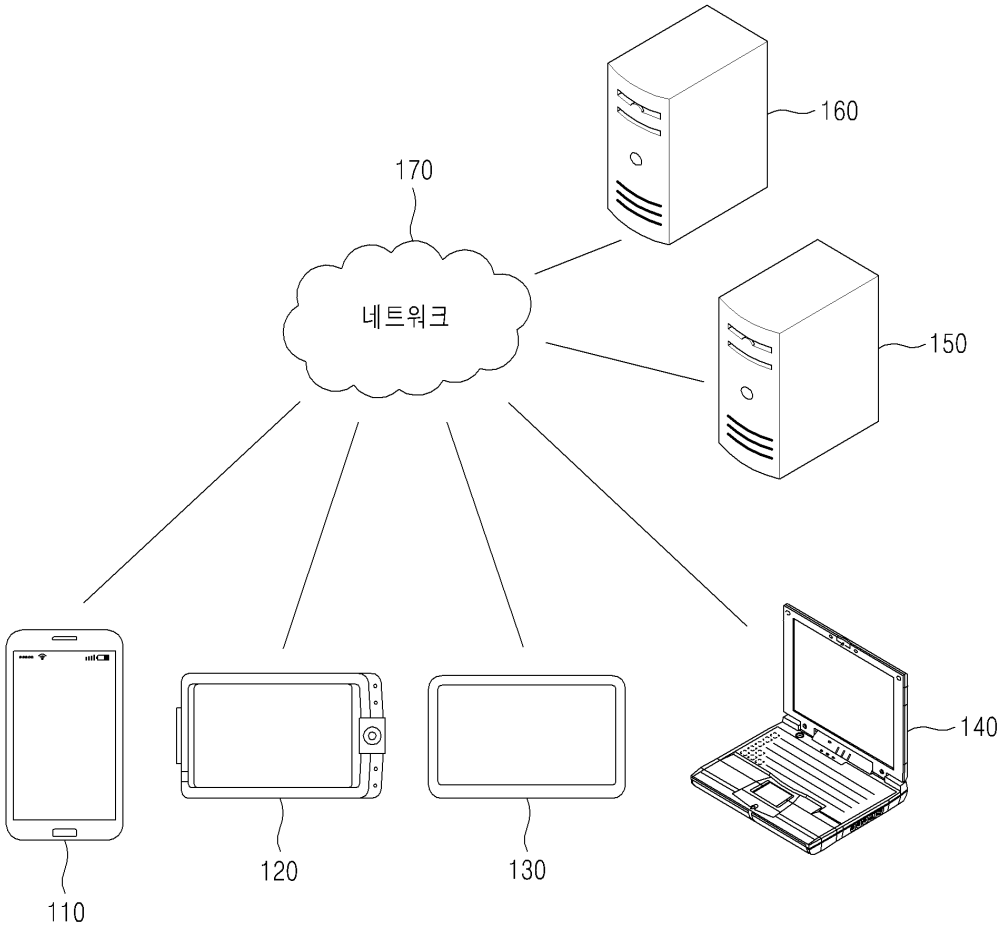
[0057] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0058] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

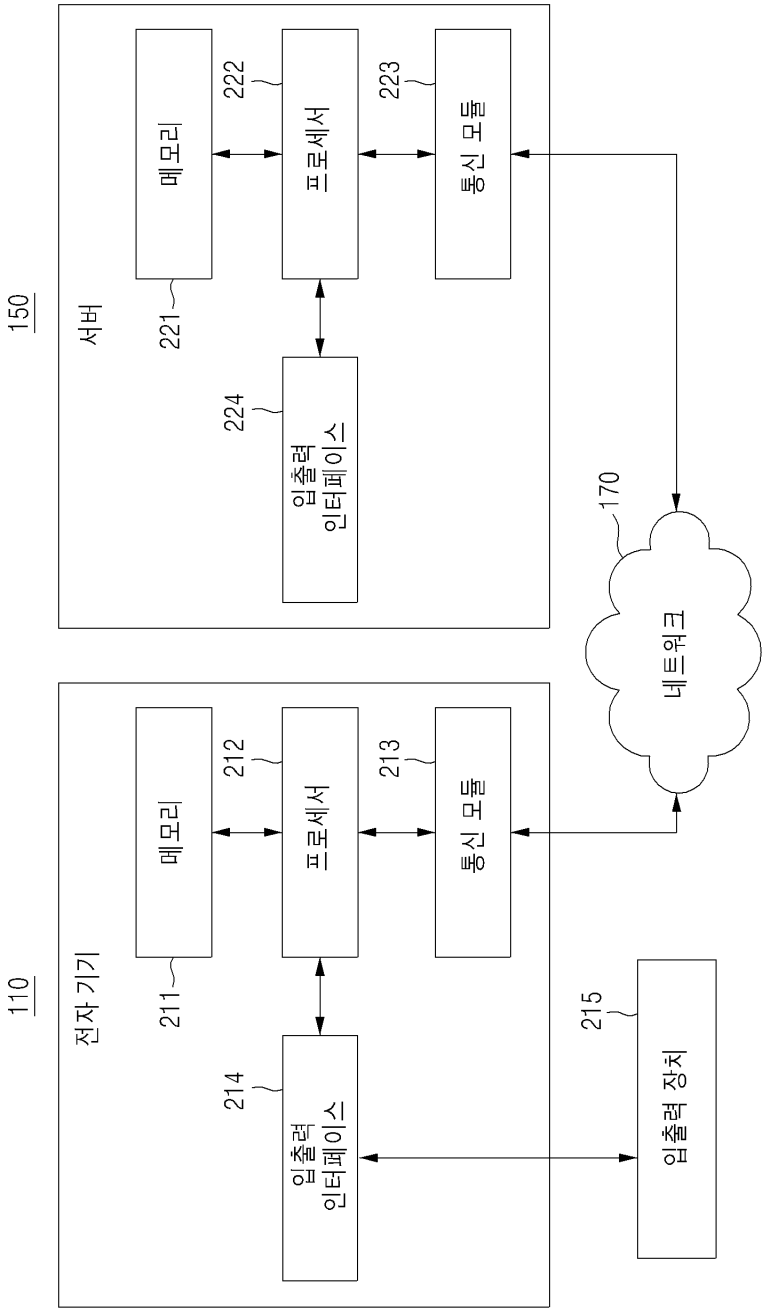
[0059] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

도면1



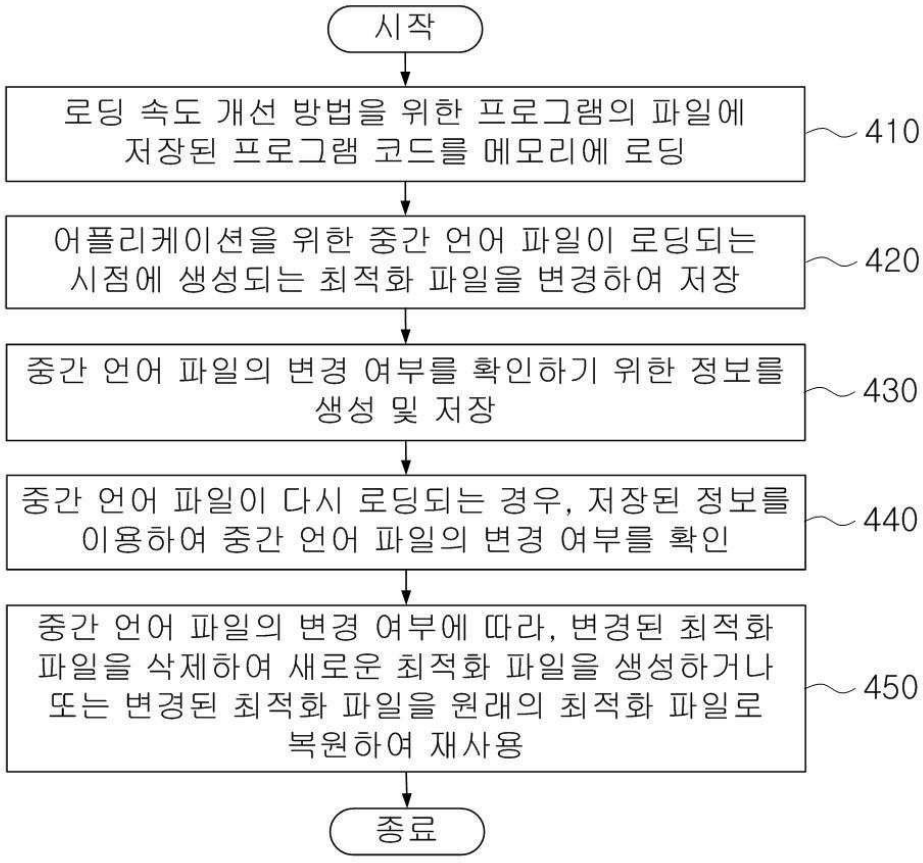
도면2


150
서버
메모리
221
222
프로세서
223
통신 모듈
224
입출력
인터페이스
170
네트워크
110
전자 기기
211
메모리
212
프로세서
213
통신 모듈
214
입출력
인터페이스
215
입출력 장치

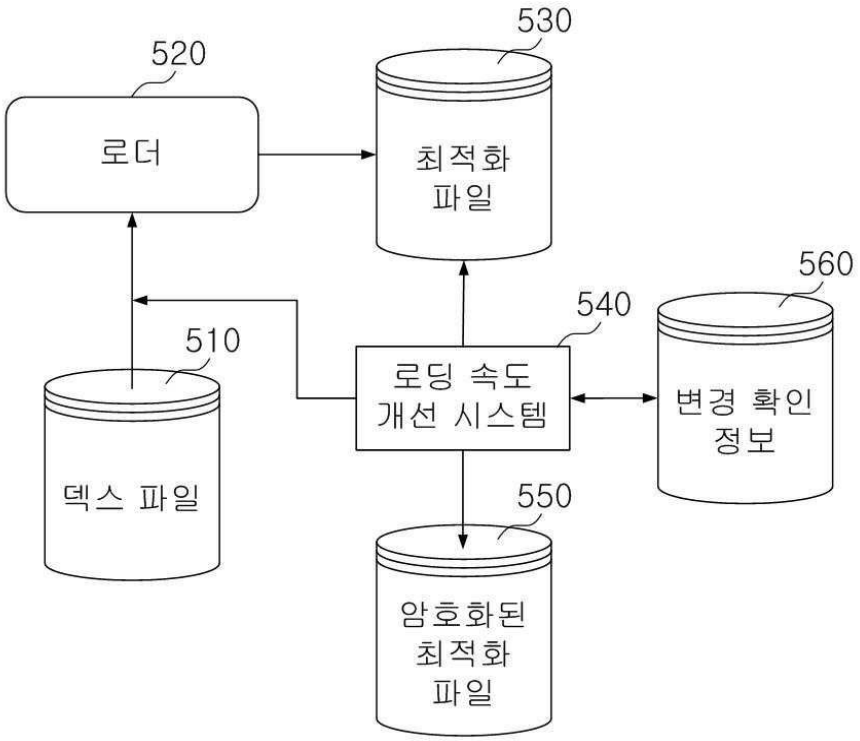
도면3


212
프로세서
310
로딩 제어부
320
최적화 파일 관리부
330
변경 확인 정보 관리부
340
변경 확인부
350
최적화 파일 재사용 제어부

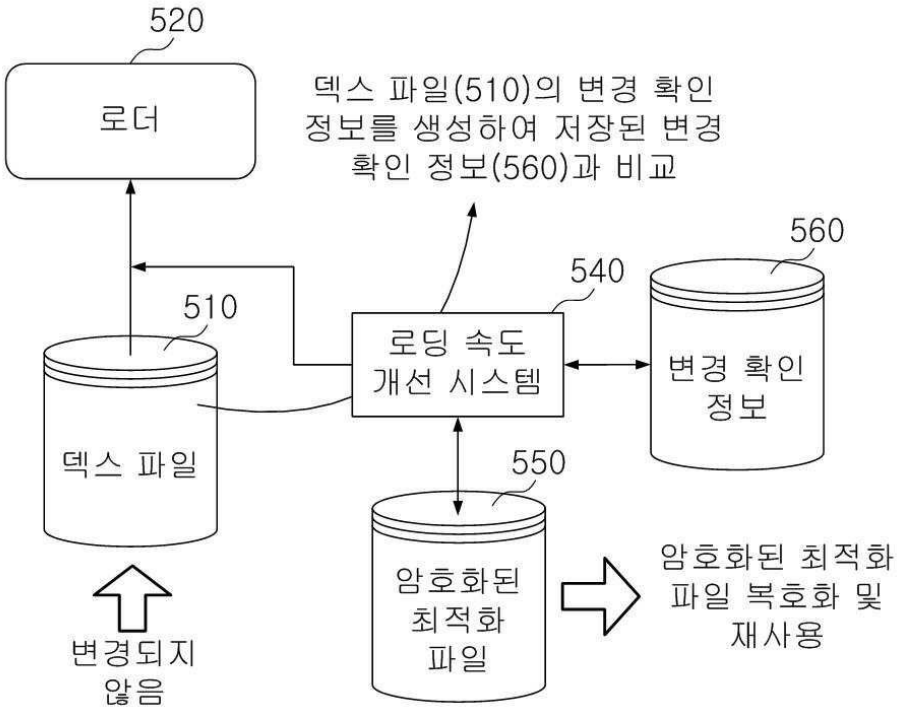
도면4


시작
로딩 속도 개선 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩
410
어플리케이션을 위한 중간 언어 파일이 로딩되는 시점에 생성되는 최적화 파일을 변경하여 저장
420
중간 언어 파일의 변경 여부를 확인하기 위한 정보를 생성 및 저장
430
중간 언어 파일이 다시 로딩되는 경우, 저장된 정보를 이용하여 중간 언어 파일의 변경 여부를 확인
440
중간 언어 파일의 변경 여부에 따라, 변경된 최적화 파일을 삭제하여 새로운 최적화 파일을 생성하거나 또는 변경된 최적화 파일을 원래의 최적화 파일로 복원하여 재사용
450
종료

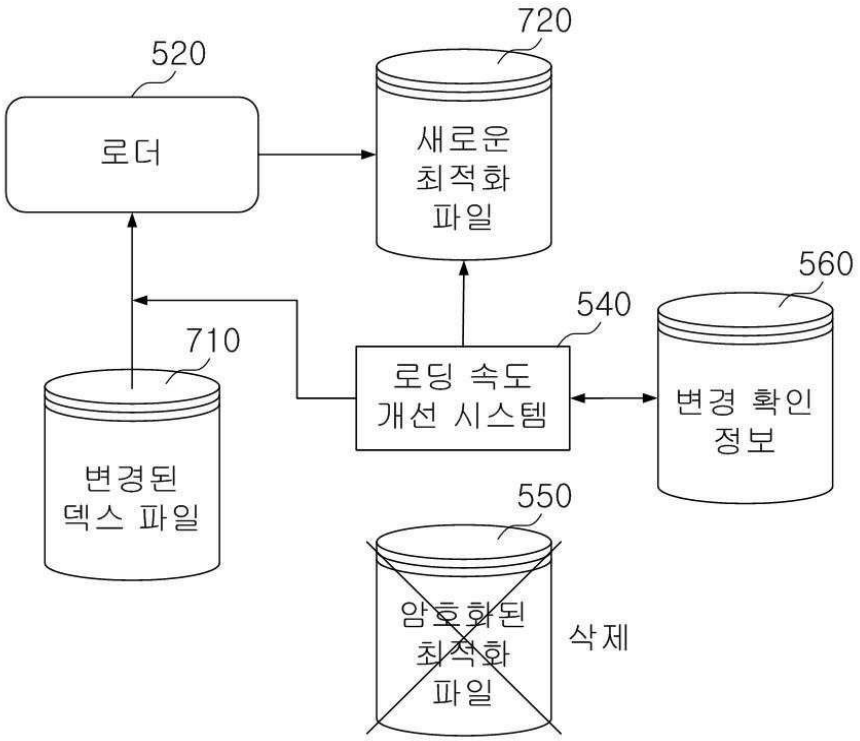
도면5


520
로더
530
최적화
파일
560
540
510
로딩 속도
개선 시스템
변경 확인
정보
덱스 파일
550
암호화된
최적화
파일

도면6


520
로더
덱스 파일(510)의 변경 확인
정보를 생성하여 저장된 변경
확인 정보(560)과 비교
560
540
510
로딩 속도
개선 시스템
변경 확인
정보
덱스 파일
550
암호화된
최적화
파일
암호화된 최적화
파일 복호화 및
재사용
변경되지
않음

도면7


520
로더
720
새로운
최적화
파일
560
540
710
로딩 속도
개선 시스템
변경 확인
정보
변경된
덱스 파일
550
암호화된
최적화
파일
삭제