



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년06월18일
 (11) 등록번호 10-1530104
 (24) 등록일자 2015년06월12일

(51) 국제특허분류(Int. Cl.)
 G06F 9/24 (2006.01) G06F 9/445 (2006.01)
 H04B 1/40 (2015.01)
 (21) 출원번호 10-2010-7006351
 (22) 출원일자(국제) 2008년09월17일
 심사청구일자 2013년08월13일
 (85) 번역문제출일자 2010년03월23일
 (65) 공개번호 10-2010-0058583
 (43) 공개일자 2010년06월03일
 (86) 국제출원번호 PCT/US2008/076619
 (87) 국제공개번호 WO 2009/045717
 국제공개일자 2009년04월09일
 (30) 우선권주장
 11/864,648 2007년09월28일 미국(US)
 (56) 선행기술조사문헌
 KR1020060124047 A
 US20030093658 A1

(73) 특허권자
마이크로소프트 코포레이션
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
 마이크로소프트 웨이
 (72) 발명자
바리볼트, 그레고리 피.
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로
 소프트 웨이 마이크로소프트 코포레이션 국제 특
 허부 내
모리스, 래리 에이.
 미국 98052-6399 워싱턴주 레드몬드 원 마이크로
 소프트 웨이 마이크로소프트 코포레이션 국제 특
 허부 내
 (뒷면에 계속)
 (74) 대리인
김태홍

전체 청구항 수 : 총 15 항

심사관 : 지정훈

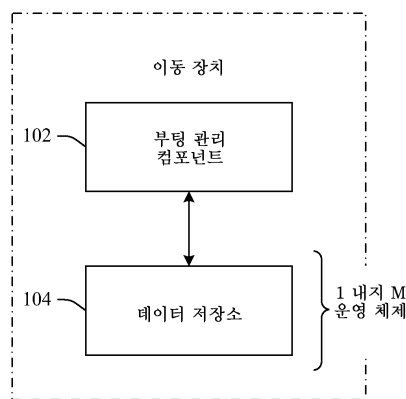
(54) 발명의 명칭 **모바일 장치를 통한 다중-OS(운영 체제) 부팅**

(57) 요약

사용자들에게 특정한 모바일 장치 기능들, 보안, 애플리케이션 등에서의 신속한 액세스를 제공하도록 운영 체제(OS)의 상이한 레벨들을 용이하게 하는 시스템이 제공된다. 정책들, 선호들, 액션들, 컨텍스트 등과 잘 맞는 국면들 또는 스테이지들에서 OS가 로딩될 수 있다. 유사하게, 특정한 시나리오 또는 요건을 다루기 위해 특정한 OS 또는 OS들의 그룹이 로딩될 수 있다.

대표도 - 도1

100



(72) 발명자

구데이, 샤이

미국 98052-6399 워싱턴주 레드몬드 윈 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

밀러, 존 마크

미국 98052-6399 워싱턴주 레드몬드 윈 마이크로소프트 웨이 마이크로소프트 코포레이션 국제 특허부 내

명세서

청구범위

청구항 1

모바일 셀룰러 폰(mobile cellular telephone)으로서,
프로세싱 하드웨어; 및

상기 프로세싱 하드웨어와 연결되며, 하이퍼바이저(hypervisor) 및 적어도 제1 OS(operating system) 및 제2 OS를 포함하는 복수의 OS들을 저장하는 데이터 스토리지 하드웨어

를 포함하고,

상기 프로세싱 하드웨어는, 상기 모바일 셀룰러 폰이 작동하고 있을 때, 상기 데이터 스토리지 하드웨어와 협력하여 상기 하이퍼바이저를 실행시키며, 상기 하이퍼바이저는, 실행되고 있을 때에, 상기 제1 OS와 상기 제2 OS를 동시에(concurrently) 부팅하고 실행시키기 위해 머신 가상화(machine virtualization)를 제공하고, 상기 제1 OS는 상기 제2 OS보다 짧은 부팅 시간을 가지며, 상기 제1 OS는, 상기 제1 OS가 부팅을 완료했을 때에 그리고 상기 제2 OS가 부팅을 완료하기 전에, 사용자에게 의한 사용이 가능해지는 셀룰러 음성 소프트웨어를 포함하고, 상기 셀룰러 음성 소프트웨어는 상기 제2 OS가 부팅을 계속하는 동안 사용가능하며, 상기 하이퍼바이저는 상기 제1 OS가 부팅을 완료한 후에 그리고 상기 제2 OS가 부팅을 완료한 후에, 상기 제1 OS와 상기 제2 OS를 동시에 실행시키는 것인, 모바일 셀룰러 폰.

청구항 2

제1항에 있어서, 상기 데이터 스토리지 하드웨어의 적어도 일부는 클라우드 기반 컴포넌트(cloud-based component)를 포함하는 것인, 모바일 셀룰러 폰.

청구항 3

제1항에 있어서, 상기 모바일 셀룰러 폰이 파워-온되었을(powered-on) 때, 상기 하이퍼바이저는 상기 프로세싱 하드웨어에 의해 실행되어 상기 하이퍼바이저에 의해 부팅하기 위한 상기 복수의 OS들 중 적어도 하나의 OS를 선택하는 것인, 모바일 셀룰러 폰.

청구항 4

제3항에 있어서, 상기 데이터 스토리지 하드웨어는 상기 데이터 스토리지 하드웨어의 파티션에서 부트 섹터(boot sector)를 식별하고 액세스하기 위해 마스터 부팅 코드 및 파티션 테이블을 이용하도록(employ) 구성된 마스터 부팅 레코드를 더 포함하는 것인, 모바일 셀룰러 폰.

청구항 5

제3항에 있어서, 부팅 관리 컴포넌트는 상기 복수의 OS들 중 적어도 하나의 OS를 선택하기 위한 컨텍스트(context)를 이용하는(employ) 컨텍스트 기반 선택 컴포넌트(context-based selection component)를 포함하는 것인, 모바일 셀룰러 폰.

청구항 6

제5항에 있어서, 상기 셀룰러 폰이 작동하고 있을 때에, 상기 컨텍스트를 확립하도록(establish) 센싱된 데이터(sensed data)를 제공하는 복수의 센서들을 더 포함하는 것인, 모바일 셀룰러 폰.

청구항 7

제6항에 있어서, 상기 데이터 스토리지 하드웨어는, 상기 OS들 중 적어도 하나의 OS의 섹션을 제어하는 정책을 더 저장하며, 상기 정책은 상기 센싱된 데이터에 따라서 적용되는 것인, 모바일 셀룰러 폰.

청구항 8

제1항에 있어서, 상기 데이터 스토리지 하드웨어는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 정의된 기준(a defined criterion)에 기초하여 상기 복수의 OS들 중 적어도 하나의 OS를 선택하도록 상기 프로세싱 하드웨어에 의해 실행되는 규칙 기반 선택 컴포넌트(rules-based selection component)를 더 저장하는 것인, 모바일 셀룰러 폰.

청구항 9

제1항에 있어서, 상기 제1 OS는 음성 명령(voice command)의 작동을 가능하게 하도록 구성되는 것인, 모바일 셀룰러 폰.

청구항 10

제1항에 있어서, 상기 데이터 스토리지 하드웨어는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 상기 모바일 셀룰러 폰이 다른 사용자에게 핸드오프되고 있다는 결정을 나타내는 신호에 응답하여 상기 복수의 OS들 중 적어도 하나의 OS의 기능을 제한하도록, 상기 프로세싱 하드웨어에 의해 실행되는 핸드오프 모드 컴포넌트(hand-off mode component)를 더 저장하는 것인, 모바일 셀룰러 폰.

청구항 11

제1항에 있어서, 상기 데이터 스토리지 하드웨어는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 상기 모바일 셀룰러 폰을 작동시키는 사람의 아이덴티티(identity)를 결정하는 함수(function)로서 상기 복수의 OS들 중 적어도 하나의 OS의 선택을 가능하게(enable) 하도록, 상기 프로세싱 하드웨어에 의해 실행되는 아이덴티티 기반 선택 컴포넌트(identity-based selection component)를 더 저장하는 것인, 모바일 셀룰러 폰.

청구항 12

제1항에 있어서, 상기 데이터 스토리지 하드웨어는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 상기 데이터 스토리지 하드웨어를 복수의 파티션들로 파티셔닝하고, 상기 복수의 파티션들 내의 상기 복수의 OS들 중 적어도 하나의 OS를 찾아내고(locate), 상기 복수의 OS들 중 상기 적어도 하나의 OS의 시작(launch)을 가능하게 하도록, 상기 프로세싱 하드웨어에 의해 실행되는 구현 컴포넌트(implementation component)를 더 저장하는 것인, 모바일 셀룰러 폰.

청구항 13

제12항에 있어서, 상기 파티션들 각각은 상기 복수의 OS들 중 상기 적어도 하나의 OS를 시작하도록 파티션-특유의 부트 섹터(partition-specific boot sector)에 액세스하기 위하여 마스터 부팅 코드 및 파티션 테이블을 이용하도록(employ) 구성되는 것인, 모바일 셀룰러 폰.

청구항 14

모바일 셀룰러 폰으로서,

프로세서; 및

상기 프로세서와 협력하도록 구성되는 데이터 저장 컴포넌트(data stor component) - 상기 데이터 저장 컴포넌트는 적어도 제1 OS와 제2 OS를 포함하는 복수의 OS들을 저장하고, 상기 데이터 저장 컴포넌트는 상기 모바일 셀룰러 폰 상에서 상기 제1 OS와 상기 제2 OS의 동시 실행을 가능하게 하기 위한 하드웨어 가상화를 제공하는 가상화 컴포넌트를 더 저장함 -

를 포함하고,

상기 프로세서는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 상기 가상화 컴포넌트를 실행시키며, 상기 가상화 컴포넌트는 상기 데이터 저장 컴포넌트로부터 상기 제1 OS와 상기 제2 OS를 동시에 부팅시키고, 상기 제1 OS는 상기 제2 OS가 부팅을 계속하고 있는 동안 사용자에게 의한 사용이 가능해지는 애플리케이션 소프트웨어를 포함하고, 상기 제1 OS와 상기 제2 OS는 상기 제1 및 제2 OS 둘다 부팅을 완료했을 때에 상기 가상화 컴포넌트에 의해 동시에 실행되며,

하이퍼바이저는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 모바일 셀룰러 폰을 통해, 상기 복수의 OS들 중 적어도 하나의 OS를 선택하고 시작하도록 상기 프로세서에 의해 실행되고, 상기 하이퍼바이저는, 컨텍스트 기반

선택 컴포넌트를 포함하고, 임의의 사용자 정의된 규칙(user-defined rules)을 평가할 수 있는 규칙 로직(rules logic)을 포함하고, 상기 규칙 로직을 사용하여 상기 모바일 셀룰러 폰의 현재 컨텍스트에 대하여(against) 상기 데이터 저장 컴포넌트에 저장된 현재 사용자 정의된 규칙을 평가하고, 평가의 결과를 사용하여 상기 복수의 OS들 중에서 상기 제1 OS를 선택하는 것인, 모바일 셀룰러 폰.

청구항 15

모바일 셀룰러 폰으로서,

다중 운영 시스템(operating system; OS)들 - 상기 다중 OS들은 제1 OS와 제2 OS를 포함하고, 상기 제1 OS는, 실행될 때, 상기 제1 OS가 부팅을 완료한 후에 상기 모바일 셀룰러 폰에 의한 셀룰러 통신을 가능하게 하는 소프트웨어를 포함함 - 을 저장하는 데이터 저장 컴포넌트를 포함하고,

상기 데이터 저장 컴포넌트는, 상기 모바일 셀룰러 폰이 작동하고 있을 때에, 상기 제1 OS 및 상기 제2 OS가 모두 부팅을 완료하면, 상기 제1 OS 및 상기 제2 OS를 상기 모바일 셀룰러 폰 상에서 동시에 실행시키는 것을 가능하게 하기 위해, 상기 제1 OS에 대한 그리고 상기 제2 OS에 대한 하드웨어 가상화를 제공하도록 프로세서에 의해 실행되는 하이퍼바이저를 더 저장하며,

상기 셀룰러 통신은 상기 제2 OS가 부팅을 완료하고 실행 중인 동안에 상기 제1 OS 상에서 계속하여 가능한 것인, 모바일 셀룰러 폰.

발명의 설명

배경 기술

[0001] 퍼스널 컴퓨터(PC) 세계에서, 종종 사용자는 시동 또는 부팅할 때 특정한 운영 체제(OS)를 선택하는 능력을 원한다. 이 다중-OS 특징은 특정 애플리케이션들의 기능으로서 또는 단순히 사용자 선호로서 요구될 수 있다. 바꾸어 말하면, 특정한 컴퓨터 애플리케이션을 실행하기 위해 특정한 OS가 요구되는 상황들이 종종 있다. 또한, 사용자들은 특정한 OS를 이용하는 것에 매우 숙달되기 때문에, 종종 특정한 OS의 특징들, 기능들 및 이의들에 대한 강한 선호가 있다.

[0002] 오늘날, PC는 시동할 때 특정한 또는 선호 OS에 관하여 사용자에게 프롬프트(prompt)하도록 구성될 수 있다. 선택은 동일한 OS의 상이한 버전들로부터 완전히 상이한 OS들, 예를 들면, 상이한 제조사들로부터의 OS들까지 다양할 수 있다. 종종, 특정한 PC의 다수의 사용자들은 OS 특징들에 관하여 상이한 선호들을 가질 것이다. 이들 다중-부팅(multi-boot) 옵션들은 특정한 OS의 선택을 가능하게 함으로써 사용자들이 그들의 컴퓨팅 세션을 개인화할 수 있게 한다.

[0003] OS는 종종 막대한 양의 메모리 및 처리 능력을 요구하는 단일 애플리케이션(monolithic application)이다. 많은 PC들 및 노트북 컴퓨터들은 OS들의 끊임없이 성장하는 기능을 이용하기 위해 필요한 메모리 및 처리 능력이 가능하지만, 많은 모바일 장치들은 오늘날의 OS들을 처리할 능력이 없다. 개인용 모바일 장치들, 예를 들면, 휴대폰, 스마트폰, PDA(personal digital assistant), 개인용 음악 플레이어, 랩톱 등의 끊임없이 증가하는 인기로 인해, '이동성(mobility)'은 무선 공급자들의 서비스들뿐만 아니라 많은 소비자 제품들의 초점이 되었다. 예를 들면, 통신 업계에서, 소비자들은 통신 및 컴퓨팅 요구에 관하여 더 이상 장소에 의해 제한되지 않기 때문에 '이동성'은 선두(forefront)에 있다.

[0004] 모바일 장치 인기가 증가함에 따라, 전화 통화를 하고, 전자 메일에 액세스하고, 인스턴트 메시지(IM)를 통해 통신하고, 표준 오피스 타입 애플리케이션들을 이용하여 계산하고 임의의 장소로부터 온라인 서비스들에 액세스하는 능력도 계속해서 발전하였다. 이 기능의 증가와 함께 증가된 기능을 처리하는 보다 크고 보다 포괄적인 OS들이 요구되고 있다. 유감스럽게도, 이들 모바일 장치들은 항상 오늘날의 OS의 요건들이 가능한 것은 아니다. 더 상세하게는, 소비자가 오늘날 스마트폰을 부팅하는 대기 시간에 실망하게 되는 것은 드문 일이 아니다.

발명의 내용

[0005] 다음은 본 발명의 일부 양태들에 대한 기초적인 이해를 제공하기 위하여 본 발명의 간략화된 개요를 제공한다. 이 개요는 본 발명의 광범위 개관을 아니다. 이것은 본 발명의 중요한/결정적인 요소들을 식별하거나 또는 본 발명의 범위를 서술하기 위한 것이 아니다. 이것의 유일한 목적은 뒤에 제공되는 보다 상세한 설명의 서문으로

서 간략화된 형태로 본 발명의 일부 개념들을 제공하기 위한 것이다.

- [0006] 여기에 개시되고 청구된 발명은, 그의 일 양태에서, 사용자들에게 특정한 기능들, 보안, 애플리케이션 등에의 신속한 액세스를 제공하도록 운영 체제(OS)의 상이한 레벨들을 용이하게 하는 시스템을 포함한다. 따라서, 정책들, 신호들, 액션들, 컨텍스트 등과 잘 맞는 국면들 또는 스테이지들에서 OS가 로딩될 수 있다. 유사하게, 특정한 시나리오 또는 요건을 다루기 위해 특정한 OS 또는 OS들의 그룹이 로딩될 수 있다.
- [0007] 본질적으로, 본 발명은 논리적인 OS 선택을 가능하게 함으로써 휴대용 컴퓨팅 장치들의 끊임없이 증가하는 능력들, 저장, 및 기능을 다룬다. 바꾸어 말하면, 워드 프로세서, 브라우저, 미디어 플레이어, 네트워크 애플리케이션 등과 같은 정교한 애플리케이션들의 실행을 가능하게 하기 위해 강건한(robust) OS들이 설치될 수 있다. 또한, 원한다면, 강건한 OS 부팅 프로세스가 실행되는 것을 기다리는 것보다는 원하는 기능(예를 들면, 스마트폰의 다이얼러(dialer))에의 빠른 액세스를 용이하게 하기 위해 보다 제한된 OS가 설치될 수 있다.
- [0008] 또한, 상이한 인증 레벨들이 각각의 부팅 레벨들과 관련될 수 있다. 이들 인증 레벨들은 데이터 및/또는 기능들에의 액세스를 제한함으로써 보안을 강화하기 위해 이용될 수 있다. 예를 들면, 단순 전화를 거는 것은 인증을 전혀 또는 거의 필요로 하지 않을 수 있지만, 안전한 네트워크로의 VPN(virtual private network) 포털에의 액세스는 긴 부팅 시간 및 높은 레벨의 인증을 필요로 할 수 있다.
- [0009] 원하는 부팅 레벨을 지정하기 위해 각종의 메커니즘들(예를 들면, 음성 명령, 빠른 버튼(quick buttons), 드롭다운 메뉴 등)이 이용될 수 있다. 특정 기능들을 이용 가능하게 하기 위해 빠른 부팅이 선택된다면, 다양한 부팅 레벨들이 달성될 때 각각의 기능들을 용이하게 하기 위해 배경에서 풀 레벨 부팅(full level boot)이 동시에 일어날 수 있다.
- [0010] 또 다른 실시예는 빠른 적용 모드(quick appliance mode)를 제공한다. 예를 들면, 장치가 풀 모드(예를 들면, 강건한 OS)에 있는 경우, 그것은 누군가(예를 들면, 어린이 또는 낯선 사람)에게 넘겨져서 빠르게 게임, 또는 지역 통화만을 허용하는 상이한 모드(예를 들면, 덜 강건한(less robust) OS)로 설정될 수 있다. 이 예에서, 제한된 OS는 기밀 정보 또는 기능의 액세스를 제한(또는 금지)함으로써 보안을 강화할 수 있다.
- [0011] 그의 다른 양태들에서, 사용자를 대신하여 기능을 자동화하는 규칙 기반 및 기계 학습 및 추론(MLR; machine learning & reasoning) 로직이 제공된다. 예를 들면, MLR 로직은 사용자가 자동으로 수행되기를 원하는 액션을 예지하거나 추론하기 위해 확률 및/또는 통계 기반 분석을 이용할 수 있다.
- [0012] 전술한 및 관련 목적들을 달성하기 위해, 여기서는 다음의 설명 및 첨부 도면들과 관련하여 본 발명의 특정한 예시적인 양태들이 설명된다. 그러나, 이들 양태들은 본 발명의 원리들이 이용될 수 있는 다양한 방식들 중 소수만을 나타내고 본 발명은 모든 그러한 양태들 및 그들의 균등물들을 포함하도록 의도되어 있다. 본 발명의 다른 이점들 및 신규한 특징들은 도면들과 함께 고려될 때 본 발명에 대한 다음의 상세한 설명으로부터 명백해질 것이다.

도면의 간단한 설명

- [0013] 도 1은 본 발명의 양태에 따른 운영 체제(OS) 부팅 레벨을 관리하는 시스템을 예시한다.
- 도 2는 본 발명의 양태에 따른 다중-스테이지 OS 부팅을 용이하게 하는 절차들의 흐름도의 예를 예시한다.
- 도 3은 본 발명의 양태에 따른 모바일 장치로의 다중-OS 설치를 용이하게 하는 절차들의 흐름도의 예를 예시한다.
- 도 4는 본 발명의 양태에 따른 다중-OS 가능한 모바일 장치에 따른 부팅 관리 컴포넌트 및 파티셔닝된 데이터 저장소의 블록도의 예를 예시한다.
- 도 5는 본 발명의 양태에 따른 데이터 저장소 파티션 구조의 예를 예시한다.
- 도 6은 본 발명의 양태에 따른 OS 선택 및 구현을 용이하게 하는 부팅 관리 컴포넌트의 블록도의 예를 예시한다.
- 도 7은 본 발명의 양태에 따른 컨텍스트 기반 및 아이덴티티(identity) 기반 선택을 가능하게 하는 선택 컴포넌트의 블록도의 예를 예시한다.
- 도 8은 본 발명의 양태에 따른 선택된 OS(또는 OS의 스테이지)를 발견하고 로딩하는 구현 컴포넌트의 블록도의

예를 예시한다.

도 9는 본 발명의 양태에 따른 다중-스테이지 또는 다중-OS 부팅이 가능한 모바일 장치의 블록 아키텍처 도를 예시한다.

도 10은 개시된 아키텍처를 실행하도록 동작 가능한 컴퓨터의 블록도를 예시한다.

도 11은 본 발명에 따른 예시적인 컴퓨팅 환경의 개략 블록도를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0014] 지금부터 첨부 도면들을 참조하여 본 발명을 설명하는데, 첨부 도면들에서, 유사한 참조 번호들은 전체에 걸쳐서 유사한 엘리먼트들을 지시하기 위해 이용된다. 다음의 설명에서는, 설명의 목적으로, 본 발명의 철저한 이해를 제공하기 위해 다수의 특정 상세들이 제시된다. 그러나, 본 발명은 이들 상세들 없이도 실시될 수 있다는 것이 명백할 것이다. 다른 사례들에서, 본 발명의 설명을 용이하게 하기 위하여 잘 알려진 구조들 및 장치들은 블록도 형태로 제시된다.

[0015] 이 출원에서 사용될 때, "컴포넌트" 및 "시스템"이라는 용어들은 하드웨어이든, 하드웨어와 소프트웨어의 조합이든, 소프트웨어이든, 또는 실행 중인 소프트웨어이든 간에, 컴퓨터 관련 엔티티를 지시하기 위한 것이다. 예를 들면, 컴포넌트는 프로세서에서 실행하는 프로세스, 프로세서, 개체, 실행 파일(executable), 실행 스레드(thread of execution), 프로그램, 및/또는 컴퓨터일 수 있지만, 그것에 제한되지 않는다. 예로서, 서버에서 실행하는 애플리케이션 및 서버는 양쪽 모두 컴포넌트일 수 있다. 하나 이상의 컴포넌트들이 프로세스 및/또는 실행 스레드 내에 존재할 수 있고, 컴포넌트는 하나의 컴퓨터 상에 국한(localize)되거나 및/또는 2개 이상의 컴포넌트들 사이에 분산될 수 있다.

[0016] 여기에서 사용될 때, "추론하다(infer)" 또는 "추론"이라는 용어는 일반적으로 이벤트들 및/또는 데이터를 통해 캡처된 관찰들의 세트로부터 시스템, 환경, 및/또는 사용자의 상태들에 관하여 추론하거나 그 상태들을 추론하는 과정을 지시한다. 추론은 특정 컨텍스트 또는 액션을 식별하기 위해 이용될 수 있고, 또는 예를 들면, 상태들에 대한 확률 분포를 생성할 수 있다. 추론은 개연론(probabilistic) - 즉, 데이터 및 이벤트들의 고려에 기초한 흥미 있는 상태들에 대한 확률 분포의 계산일 수 있다. 추론은 또한 이벤트들 및/또는 데이터의 세트로부터 보다 높은 레벨의 이벤트들을 구성하기 위해 이용되는 기법들을 지시할 수 있다. 그러한 추론은, 이벤트들이 시간적으로 밀접하게 상관되어 있든 아니든 간에, 그 이벤트들 및 데이터가 하나의 이벤트 및 데이터 소스로부터 유래한 것이든 몇 개의 이벤트 및 데이터 소스들로부터 유래한 것이든 간에, 관찰된 이벤트들 및/또는 저장된 이벤트 데이터의 세트로부터의 새로운 이벤트들 또는 액션들의 구성으로 귀결한다.

[0017] 휴대용 컴퓨팅 장치들(예를 들면, 스마트폰들)은 처리 능력들, 저장, 및 기능에서 계속적으로 증가한다. 그 결과, 사용자들은 워드 프로세서, 인터넷 브라우저 및 미디어 플레이어와 같은 정교한 애플리케이션들을 설치 및 실행한다. 또한, 핵심 기능들(예를 들면, 스마트폰의 다이얼러) 외에도, 많은 모바일 장치들은 오늘날 안전한 네트워크들에 액세스하기 위해 이용된다. 보다 많은 양의 애플리케이션 사용의 결과로서, 이들 장치들의 능력들을 다루기 위해 보다 강건한 운영 체제들(OS들)이 요구되기 때문에 그러한 장치들에 대한 부팅 시간은 더 많이 걸린다.

[0018] 우선 도면들을 참조하여, 도 1은 부팅 관리 컴포넌트(102) 및 데이터 저장소를 그 안에 갖는 모바일 장치(100)를 예시한다. 동작시에, 부팅 관리 컴포넌트(102)는 모바일 장치(100)를 통해 다중-부팅 또는 다중-로그온 기능을 가능하게 할 수 있다. 본질적으로, OS 부팅을 관리함으로써, 부팅 관리 컴포넌트(102)는 아이덴티티(예를 들면, 사람), 기업 연합, 컨텍스트, 정책, 선호 등을 포함하지만 이들에 제한되지 않는 거의 임의의 요인들에 따라서 기능 및 애플리케이션들을 이용 가능하게 할 수 있다.

[0019] 예로서, 부팅 관리 컴포넌트(102)는 사용자에게 신속히 기능을 제공하기 위하여 스트림라인 OS를 로딩할 수 있다. 한편, 컴포넌트(102)는 추가적인 특징들 및 기능들을 이용 가능하게 하기 위하여 OS의 강건한 또는 완전한 버전을 계속해서 로딩할 수 있다. 또 다른 실시예들에서, OS의 버전들은 '아이덴티티', 컨텍스트, 정책 등에 기초하여 로딩될 수 있다. 이 다중-스테이지 부팅 또는 로그온 절차는 원하는 대로 거의 임의의 방식으로 사용자 지정(customize)되거나 변경될 수 있다.

[0020] 전술한 바와 같이, 부팅 관리 컴포넌트(102)는 최소한도의 기능 상태가 신속히 도달될 수 있게 할 수 있는 반면 다른 기능은 완전한 OS 설치시에 이용 가능하게 된다. 예로서, 모바일 장치(100)는 배경에서 보다 강건한(또는 상이한) OS를 계속해서 부팅하면서 사용자가 긴급 통화 또는 원하는 통화를 하기 위해 다이얼러를 이용가능하게

끔 할 수 있다. 이 예에서는, 메모리로부터 콘택트 정보가 입력되기만 하면 사용자가 전화 통화를 하는 것이 가능할 수 있다. 바꾸어 말하면, 다중-스테이지 부팅의 일부는 사용자가 기업 콘택트들(enterprise contacts)에 액세스할 수 있을 '기업 스테이지 OS'를 지시할 수 있다. 여기서, 이 스테이지는 액세스 및/또는 허가(authorization)를 실현하기 위해 보다 강건한 OS를 요구할 수 있다.

[0021] 본질적으로, 부팅 관리 컴포넌트(102)는 하이퍼바이저(hypervisor)로 칭해질 수 있고 가상화 기술에서, '하이퍼바이저'는 단일 컴퓨팅 플랫폼 상의 다수의 OS들(또는 동일한 OS의 다수의 인스턴스들)을 관리하는 소프트웨어 애플리케이션이다. 하이퍼바이저는 각 OS가 요구하는 것을 할당하기 위해 시스템의 메모리, 프로세서 및 다른 리소스들을 관리할 수 있다. 이 발명에 따르면, 이러한 관리는 아이덴티티 기반, 정책 기반, 컨택트 기반 등일 수 있다. 대부분의 애플리케이션들에서, 하이퍼바이저들은 특정한 프로세서 아키텍처에 대하여 설계되고 또한 가상화 관리자(virtualization managers)로 칭해질 수도 있다.

[0022] 예시된 바와 같이, 데이터 저장소 컴포넌트(104)는 1 내지 M 운영 체제를 포함할 수 있고, 여기서 M은 정수이다. 양태들에서, 이들 OS들은 제한된 기능, 부분 기능, 전체 기능, 애플리케이션 기반, 아이덴티티 기반 등을 나타낼 수 있다. 따라서, OS들 각각은 데이터 저장소 컴포넌트(104) 내에 유지될 수 있다. 또한, 비록 데이터 저장소 컴포넌트(104)는 모바일 장치(100)에 포함되는 것으로 예시되어 있지만, 데이터 저장소 컴포넌트(104)의 전부 또는 일부는 본 발명의 정신 및/또는 범위에서 벗어나지 않고 원격에 위치할 수 있다는 것을 이해해야 한다. 예로서, 데이터 저장소 컴포넌트(104)(또는 그의 일부)는 네트워크(예를 들면, 인터넷) 접속 또는 액세스를 통해 액세스가 실행될 수 있도록 '클라우드 기반(cloud-based)'일 수 있다.

[0023] 다음에 오는 설명은 본 발명의 핵심 기능의 특징들, 기능들 및 이익들을 포함한다. 바꾸어 말하면, 부팅/로그온 스테이지들을 제어 및/또는 관리하는 기능은 본 발명의 정신 및/또는 범위에서 벗어나지 않고 거의 임의의 방식으로 실행될 수 있다. 다음의 오는 설명은 부팅 스테이지들을 선택하고, 사용자 지정하고, 용도에 맞추고(tailoring), 개인화하고 및/또는 자동화하는 다수의 예들을 설명한다 - 이들 모두는 본 발명의 일부로 간주되어야 할 것이다.

[0024] 도 2는 본 발명의 양태에 따른 다중-부팅 또는 부분-부팅의 방법의 예를 예시한다. 전술한 바와 같이, 본 발명은 모바일 장치가 특정한 스테이지(들)에 따른 기능들의 능력을 강화할 수 있도록 OS(또는 OS들의 그룹)가 스테이지들에서 부팅될 수 있게 한다. 일 양태에서, 제한된 또는 초기 스테이지는 전체 기능을 사용 가능하게 하기 전에 스마트폰의 다이얼러를 사용 가능하게 함으로써 사용자가 통화를 하는 것을 가능하게 할 수 있다. 일단 다이얼러가 사용 가능하게 되면, 스마트폰은 OS가 적절한 국면들에 도달할 때 추가적인 기능들이 이용 가능해질 수 있도록 OS의 강건한 버전을 계속해서 로딩할 수 있다.

[0025] 또한, 이 OS 스테이지-부팅은 아이덴티티, 컨택트 등과 같은 요인들에 기초하여 사용자에게 의해 트리거되거나 자동으로 트리거될 수 있다. 이들 및 다른 예들은 다음에 오는 도면들에 관련하여 더 상세히 설명될 것이다. 여기에는 많은 예들 및 실시예들이 포함되지만, 이들 예들은 단지 본 발명에 대한 관점을 추가하기 위해 제공되는 것임을 이해하고 인식해야 한다. 따라서, 이 발명의 범위 내에 포함되어야 하는 추가적인 예들이 존재한다.

[0026] 설명의 간결함을 위하여, 예를 들면, 흐름도의 형태로, 여기에 제시된 하나 이상의 방법들은 일련의 단계(act)들로서 도시되고 설명되지만, 어떤 단계들은, 본 발명에 따라, 여기에 도시되고 설명된 것과 다른 순서로 및/또는 다른 단계들과 동시에 일어날 수 있으므로, 본 발명은 단계들의 순서에 의해 제한되지 않는다. 예를 들면, 당업자들은 방법이 대안적으로는 상태도에서와 같이 일련의 상호 관련된 상태들 또는 이벤트들로서 나타내어질 수 있다는 것을 이해하고 인식할 것이다. 또한, 본 발명에 따른 방법을 구현하기 위해 예시된 모든 단계들이 다 요구되지 않을 수도 있다.

[0027] 단계(202)에서는, 이용 가능한 OS들이 식별될 수 있다. 동작시에, 모바일 장치에 이용 가능한 OS들은 모바일 장치 상에서 및/또는 네트워크 접속된 위치(예를 들면, 클라우드 기반, 인터넷, 인트라넷) 등으로부터 발견될 수 있다. 도 2의 예에서, 사용자는 단계(202)에서 단정적으로 또는 명시적으로 OS를 선택할 수 있다. 단계(204)에서는 사용자가 장치를 부팅할 때 이용할 특정한 OS를 명시적으로 또는 단정적으로 선택하는지를 결정하는 관정이 행해진다. 여기서, 특정한 OS를 선택하기 위해 음성 명령, 빠른 버튼, 드롭-다운 또는 풀-다운 메뉴 등을 포함하지만 이들에 제한되지 않는 각종의 메커니즘들이 이용될 수 있다.

[0028] 또한, 이들 메커니즘들은 부팅시에 특정한 OS를 식별하기 위해 분석될 수 있는 특정 기능들을 선택하기 위해 이용될 수 있다. 예를 들면, 사용자는 '통화를 하는(make a call)' 또는 다른 명령을 표명할 수 있고 그에 의해 장치는 사용자가 통화를 하기에 충분한 기능을 이용 가능하게 하는 소프트-부팅(또는 제한된 부팅)을 원한다는

것을 자동으로 결정할 것이다. 전술한 바와 같이, 이 시나리오에서, 장치는 전화 통화를 하는 사용자의 능력을 연기하지 않고 전체 기능을 이용 가능하게 하기 위하여 완전한 또는 강건한 OS를 계속해서 부팅할 수 있다.

[0029] 만일 단계(204)에서 사용자에게 의해 OS가 선택되지 않으면, 단계(206) 및 단계(208)에서 각각 선택이 컨텍스트 기반이어야 하는지 정책 기반이어야 하는지에 대한 결정이 행해질 수 있다. 바꾸어 말하면, 현재의 컨텍스트 또는 정책에 따라서 OS(또는 OS들의 그룹)의 선택이 자동화될 수 있다. 비록 도 2의 예는 특정 판정 로직에 관한 것이지만, 본 발명의 정신 및/또는 범위에서 벗어나지 않고 다른 유형의 로직을 이용하는 다른 예들이 존재한다는 것을 이해해야 한다. 추가 예로서, 사용자 또는 기업을 대신하여 판정을 행하기 위해 기계 학습 및/또는 추론(MLR)이 이용될 수 있다.

[0030] 먼저 단계(206)의 컨텍스트 기반 판정을 참조하여, 여기서, 적절한 OS(또는 그의 일부)를 선택하기 위해 장소, 시각(time-of-day), 움직임, 생리적 요인, 환경 요인 등과 같은 컨텍스트 인식이 이용될 수 있다. 도시된 바와 같이, 단계(206)에서 컨텍스트 기반이 적용되어야 한다면, 단계(210)에서, 예를 들면, 글로벌 위치 센서, 캘린더, 클록, 환경 센서, 생리적 센서 등과 같은 감각 메커니즘들을 통해 컨텍스트가 확립될 수 있다. 도 2의 점선에 의해 예시된 바와 같이, 컨텍스트와 함께 또는 컨텍스트에 따라서 정책이 적용될 수 있다.

[0031] 단계(208)에서는, 정책 기반 예에서, 부팅을 위한 적절한 OS(또는 그의 일부)를 선택하기 위해 규칙 또는 구현 방식(implementation scheme)이 적용될 수 있다. 바꾸어 말하면, 원하는 액션에 기초하여, 정의된 또는 미리 프로그램된 정책이 이용될 수 있다. 단계(212)에서는, 예를 들면, 내부 또는 외부 데이터 저장소로부터 정책이 액세스될 수 있다.

[0032] 단계(214)에서는 컨텍스트 및/또는 정책에 따라서 적절한 OS가 발견될 수 있다. 여기서, 사용자 선호, 액션, 또는 컨텍스트에 따라서 부팅 시간을 향상 및/또는 최적화하기 위해 OS 스테이지들이 정의될 수 있다는 것을 이해해야 한다. 일단 식별되면, OS(또는 스테이지)는 단계(216)에서 시작(launch)될 수 있다. 시스템에 의해 이용되는 로직은 본 발명의 정신 및 범위에서 벗어나지 않고 거의 임의의 미리 정의된 및/또는 추론 기반(예를 들면, MLR) 로직을 포함할 수 있다.

[0033] 이제 도 3을 참조하면, 본 발명에 따른 다수의 OS들을 설치하는 방법이 예시되어 있다. 전술한 바와 같이, 종종, 원하는 애플리케이션 또는 기능을 실행하기 위해 상이한 OS들이 선호되거나 또는 때때로 요구되기도 한다. 유사하게, 본 발명에 따르면, 특정한 OS가 스테이지들(예를 들면, 가벼운(light) 또는 강건한)로 분할될 수 있고 그에 의해 OS의 부분들 또는 스테이지들 각각이 특정한 기능 또는 기능들의 그룹을 가능하게 할 수 있다.

[0034] 이제 도 3을 참조하여, 단계(302)에서는, 선호, 시나리오, 애플리케이션, 기능 등에 따라서 복수의 OS들이 식별될 수 있다. 이 식별은 원하는 대로 자동 또는 수동일 수 있다는 것을 이해할 것이다. 단계(304)에서는 규칙들/선호들의 정책 또는 세트가 확립될 수 있다. 바꾸어 말하면, 정책은 특정한 아이덴티티(또는 사람) 등과 관련하여 특정한 시나리오, 상황, 컨텍스트에서 어느 OS를 적용(또는 실행)할지를 식별할 수 있다. 본질적으로, 단계(304)에서 확립된 정책은 거의 임의의 OS(또는 그의 일부)를 거의 임의의 시나리오, 아이덴티티, 컨텍스트, 애플리케이션 등에 관련시킬 수 있다.

[0035] 단계(306)에서는 OS들이 정책에 따라서 매핑될 수 있다. 예로서, 사용자가 OS 'A'를 긴급 통화를 하는 선택에 매핑한다고 가정한다. 여기서, 이 OS 'A'는 이 상황에 매핑될 수 있고 그에 따라 긴급 통화를 하는 선택이 (수동으로 또는 자동으로) 선택되는 때는 언제나, 통화를 하는 액션을 가능하게 하기 위해 이 OS가 선택되어 로딩될 수 있다. 전술한 바와 같이, 이 얼마간 제한된 OS는 강건한 OS가 로딩하는 것을 기다릴 필요 없이 신속히 통화가 행해질 수 있게 한다는 것을 알 것이다.

[0036] 다시 도 3을 참조하여, 단계(308)에서는, 데이터 저장소를 파티셔닝함으로써 OS들의 저장에 시작될 수 있다. 이해되는 바와 같이, 다수의 OS 로딩을 가능하게 하는 것에 따라서, 예에서, 데이터 저장소는 각 OS가 다른 파티션에 유지되도록 파티셔닝된다. 유사하게, OS들은 본 발명의 정신 및/또는 범위에서 벗어나지 않고 다른 저장소들에 유지될 수 있다는 것을 이해할 것이다. OS, OS의 위치는 물론 임의의 매핑(예를 들면 상기 단계(306)으로부터의 매핑)을 식별하기 위해 테이블 또는 다른 적합한 상호(cross) 참조가 이용될 수 있다. 마지막으로, 단계(310)에서 OS들은 데이터 저장소(또는 저장소들의 그룹)에 설치될 수 있다.

[0037] 동작시에, OS가 결정된다 - 예를 들면, 사용자 선호, 애플리케이션, 시나리오, 컨텍스트 등에 대응하는 OS가 결정된다. 일단 결정되면, OS의 위치(예를 들면, 파티션)를 결정하기 위해 상호 참조 또는 테이블이 액세스된다. 그 후, OS는 원하는 대로 로딩될 수 있다. 또한, 정책에 의해 정의될 수 있는 바와 같이, 선택된 OS를 이용 가능하게 한 후에 배경에서 추가적인 또는 더 강건한 OS가 로딩될 수 있다. 양태들에서, 이 다중-로드 또는 다중

-부팅 기능은 본 발명의 기능을 강화하기 위하여 동시에 일어날 수 있다.

- [0038] 이제 도 4를 참조하면, 시스템(100)(예를 들면, 도 1로부터의 모바일 장치)의 대안적인 블록도의 예가 도시되어 있다. 특히, 도 4는 데이터 저장소(104)가 1 내지 N 파티션으로 분리될 수 있는 것을 예시하고, 여기서 N은 정수이다. 효과적으로, 각 파티션은 전술한 바와 같이 다른 OS를 수용할 수 있다는 것을 이해할 것이다. 당업자들에 의해 이해되는 바와 같이, 이 실시예에서, 각 OS는 데이터 저장소의 지정된 파티션 내에 저장될 수 있다.
- [0039] 동작시에, 사용자는 부팅할 OS를 단정적으로 선택할 수 있고 그 특정한 OS는 특정 애플리케이션들, 데이터, 기능 등에의 액세스를 제공할 수 있다. 이해되는 바와 같이, 이 선택은 수동으로 또는 사용자를 대신하여 자동으로 일어날 수 있다. OS를 선택하는 이 능력의 하나의 유용한 양태는 사용자가 장치의 전체 기능을 희망(또는 요구)하지 않는 경우에는 감소될 수 있다. 예로서, 사용자는 긴급(또는 빠른) 전화 통화를 하려는 희망을 승인할 수 있다 - 여기서, 장치는 OS의 덜 강건한(lesser-robust) 버전을 부팅할 수 있고 그에 의해 다이얼러 기능을 신속히 제공할 수 있다.
- [0040] 또 다른 예에서, 선택은 보안 및 개인 정보(privacy)에 기초할 수 있다. 이들 실시예들에서, 사용자는 덜 갖추어진(lesser equipped) OS를 스왑(swap)하거나 선택함으로써 기능 및 데이터/애플리케이션 액세스를 본질적으로 '잠그면서(locking out)' 다른 사람과 모바일 장치를 공유할 수 있다. 유사하게, 이 특징은 모바일 장치를 잃어버리거나 도둑맞으면, 덜 강건한 OS가 자동으로 선택되어 데이터 및 애플리케이션들의 사용자의 개인 정보가 보호될 수 있도록하는, 디폴트(default)일 수 있다. 예를 들면, 모바일 장치(예를 들면, 스마트폰)가 VPN(virtual private network)에 액세스하는 기능이 갖추어진 경우에, 이 인증, 허가 및 그에 따라, 액세스를 제공하기 위해 다소 강건한 OS가 요구된다는 것을 이해할 것이다. 본 발명에 따르면, 사용의 부족 또는 다른 원하는 기준에 기초하여 OS 레벨을 자동으로 감소시킬 디폴트들(예를 들면, 시간 제한들)을 설정함으로써 VPN 액세스를 용이하게 제한하도록 디폴트 또는 "빠른 키(quick key)"가 제공될 수 있다.
- [0041] 또한, OS 선택은 컨텍스트 인식 요인들에 기초할 수 있다. 이들 예들에서는, 선택을 실행(또는 자동화)하기 위해 거의 임의의 컨텍스트 기준이 이용될 수 있다. 예를 들면, 구현할 OS의 레벨을 지능적으로 결정하기 위해 장소, 날짜, 시각, 약속된 활동, PIM(personal information manager) 날짜 등이 이용될 수 있다. 보안 관련 양태에서, 이러한 유형의 정교한 로직은 OS 레벨의 변화를 보증하는 상황을 자동으로 검출(또는 추론)할 수 있을 것이다. 도 5는 본 발명의 실시예에 따른 데이터 저장소 파티션 구조의 예를 예시한다.
- [0042] 이제 도 5를 참조하면, 본 발명의 양태에 따른 파티션 구조의 예가 도시되어 있다. 다음의 논의는 다수의 OS들에 관하여 데이터 저장소(예를 들면, 드라이브)를 파티셔닝하는 일 실시예를 설명한다. OS 설치가 시작될 때, 시스템은 사용자에게 OS를 설치할 파티션(예를 들면, D:, E:, F:)을 지정하도록 요구할 수 있다. 오늘날, 대부분의 하드 디스크들은 최대 4개만의 파티션을 가질 수 있지만, 이 발명은 거의 임의의 원하는 수의 파티션을 의도한다. 하나의 예에서, 이들 추가적인 파티션들은 실제 파티션들이다. 다른 예들에서, 추가적인 파티션들은 주 파티션(primary partitions), 확장된 파티션(extended partitions), 및 논리적 파티션들(logical partitions)의 개념에 의하여 이용될 수 있다.
- [0043] 여기서, 전통적인 파티션들은 '주 파티션'으로 불리고 - 그 중 하나는 '확장된 파티션'으로 불릴 수 있다. 이 확장된 파티션은 '논리적 파티션'으로 알려진 임의의 수의 보다 작은 파티션들을 수용할 수 있다는 점에서 전통적인 주 파티션들을 보강한다. 그러므로, 이 구성에서는, 드라이브를 더 파티셔닝하는 능력을 증대하면서 전통적인 4개의 주 파티션 제한이 유지될 수 있다.
- [0044] 도 5의 예에서는, 하나의 확장된 파티션과 함께 2개의 주 파티션만이 도시되어 있다는 것을 이해할 것이다. 도 5에 예시된 구조는 단지 본 발명에 대한 관점을 추가하기 위해 제공되고 본 발명의 범위를 어떤 식으로든 제한하려는 것이 아니라는 것을 이해해야 한다. 오히려, 이 예는 여기에 설명되고 청구된 본 발명의 범위 내에 포함되어야 하는 다수의 예시적인 구조들 중 하나일 뿐이다.
- [0045] 동작시에, OS들을 부팅하는 것은 '액티브 파티션들(active partitions)' 및 '부팅 파티션들(boot partitions)'과 관련이 있다. 액티브 파티션은 디폴트(또는 원하는/선택된) OS를 로딩하기 위한 파일들을 갖는 지정된 파티션을 나타낸다. '부팅 로더(boot loader)'는 부팅 메뉴를 디스플레이하고 요구된(또는 선택된) OS를 로딩하는 프로그램 파일이다 - 이것은 후술되는 선택 컴포넌트의 기능이다. 종종, 디스크 상의 하나의 파티션만이 액티브 파티션일 수 있다. 부팅 파티션은 OS가 실제로 설치되는 파티션이다.
- [0046] 전술한 바와 같이, 스마트폰은 다수의 태스크들을 수행하거나, 애플리케이션들을 실행하거나, 보안을 제공하거나, 기능을 스트림라인하는 등을 위해 각종의 OS들을 이용할 수 있다. 예를 들면, 하나의 OS는 핵심 폰 기능

(예를 들면, 다이얼러)에 관한 것일 수 있는 반면 다른 것은 VPN 액세스 또는 심지어 게임에 관한 것일 수도 있다. 또한, (유사한 기능(예를 들면, 스케줄링, 데이터 관리, 개인 정보)을 어드레싱 하기 위해서라도) 상이한 제조사들을 대표하는 OS들이 제공되고 설치될 수 있다.

- [0047] 파티션에 OS의 설치를 시작할 때, 시스템은 먼저 파티션을 포맷할 수 있다. '포맷(formatting)'은 본질적으로 OS 설치를 위한 파티션 상에 파일 시스템을 생성한다는 것을 이해할 것이다. 다른 실시예들에서, 시스템은 필요하거나 적절할 때 데이터 저장소 내에 파티션들을 생성하고, 포맷하고, 크기를 조정하고, 삭제할 수 있다.
- [0048] 전술한 바와 같이, 부팅 로더는 사용자가 이용 가능한 OS들의 목록으로부터 OS를 선택하는 것을 가능하게 하는 프로그램이다. 이들 부팅 로더 프로그램들은 OS를 로딩하여 실행하거나 또는 사용자가 모바일 장치 상에 설치된 이용 가능한 OS들로부터 선택할 수 있는 (아래 선택 컴포넌트에 관련하여 설명된) 부팅 메뉴를 디스플레이하기 위해 이용된다는 것을 알 것이다. 또한, 부팅 로더 프로그램의 기능은 사용자를 대신하여 관리될 수 있고 (예를 들면, MLR 메커니즘) 그에 의해 정책, 선호, 시나리오 또는 다른 컨텍스트 요인들에 기초하여 사용자를 위해 결정이 행해진다.
- [0049] 도 5에 예시된 바와 같이, OS를 로딩하는 것은 특정한 OS에 대응하는 각 특정한 파티션 내에 마스터 부팅 레코드(MBR; master boot record) 및 부팅 섹터(boot sector)를 생성하는 것을 수반한다. 이 도면을 계속 참조하여, MBR은 파티션 테이블 및 마스터 부팅 코드(MBC; master boot code)를 포함한다. 파티션 테이블은 데이터 저장소 내에 위치하는 파티션들의 발견을 가능하게 하는 구조적 및 기하학적 정보를 포함한다. 이 정보는 도시된 바와 같이 파티션 테이블 항목들 내에 유지된다.
- [0050] MBC는 OS 설치 절차들에 의해 생성되고 OS 또는 부팅 로더 프로그램의 로딩을 가능하게 하는 실행 가능 코드를 제공한다. 일 양태에서, 및 도 5에 예시된 바와 같이, MBR은 디스크의 제1 섹터 상에 있다. 그러나, MBR은 본 발명의 범위에서 벗어나지 않고 임의의 다른 섹터 상에 위치할 수 있다.
- [0051] 파티션 테이블, 및 그 안에 포함된 테이블 항목들 각각은 종종 파티션의 생성시에 생성되고 장치에 설치되는 OS(들)와 관계없다. 대개의 경우에, 관련 부팅 섹터는 디스크 상의 각 파티션의 제1 섹터 상에 있다. 액티브 파티션에서, 부팅 섹터는 디폴트 OS 또는 부팅 로더 프로그램을 로딩할 코드를 포함한다. 부팅 파티션들에서, 부팅 섹터는 적절한 OS를 로딩하기 위해 필요한 실행 가능 코드를 갖는다.
- [0052] 이것은 본 발명의 양태에 따른 다중-OS 시스템의 구조의 하나의 예에 불과하다는 것을 이해할 것이다. 여기서, 이 구조는 모바일 장치(예를 들면, 스마트폰, PDA(personal digital assistant), 휴대폰) 내에 포함되고 그 후 특정 기능, 보안 절차, 애플리케이션 등을 구현하기 위해 액세스될 수 있다. 대안적인 양태들에서, 다수의 데이터 저장소들 또는 클라우드 기반 데이터 저장소들이 이용될 수 있다는 것을 이해할 것이다. 이들 대안적인 양태들은 이 발명의 범위 내에 포함되어야 한다.
- [0053] 이제 도 6을 참조하면, 본 발명의 양태에 따른 부팅 관리 컴포넌트(102)의 예의 블록도가 도시되어 있다. 일반적으로, 부팅 관리 프로그램(102)은 선택 컴포넌트(602) 및 본질적으로 모바일 장치에 따른 OS들의 다중-부팅 또는 다중-로딩을 가능하게 하는 구현 컴포넌트(604)를 포함할 수 있다. 동작시에, 선택 컴포넌트(602)는 이용 가능한 OS들로부터 OS(또는 OS들의 그룹)의 선택을 용이하게 하는 반면, 구현 컴포넌트(604)는 선택된 OS(또는 OS들의 그룹)의 실행 또는 초기화를 확립한다.
- [0054] 전술한 바와 같이, 모바일 장치 내의 이 다중-스테이지 OS 부팅 또는 로딩 기능은 사용자가 원하는 상태에 신속히 도달하는 것을 가능하게 할 수 있다. 바꾸어 말하면, 강건한 OS의 완전한 부팅을 기다리기보다는, 본 발명은 사용자가 보다 강건한, 전체 기능 OS가 로딩하는 것을 기다릴 필요 없이 신속하게 효율적으로 원하는 기능의 달성을 용이하게 할 수 있는 OS를 선택하는 것(또는 사용자를 대신하여 OS가 선택되는 것)을 가능하게 한다. 예들에서, OS는 스테이지들에서 로딩될 수 있고 그에 의해 계층적 기능(tiered functionality)에의 액세스를 제공한다. 또한, 스테이지들이 로딩될 때, 본 발명은 사용자에게 의한 동작을 방해하지 않고 배경에서 다른 스테이지들을 계속해서 로딩할 수 있다. 이 배경 로딩은 사용자에게 심리스(seamless)할 수 있고 완료될 때 효과적으로 전체 기능을 제공할 수 있다.
- [0055] 다른 양태들에서는, 다수의 유형들(예를 들면, 다른 제조사들)의 OS들이 로딩될 수 있다. 바꾸어 말하면, 동일한 OS의 스테이지들이 아니라, 여기서는, 상이한 기능들을 제공하는 다른 OS들이 로딩될 수 있다. 예를 들면, 이들 상이한 OS들은 상이한 애플리케이션들 및 기능들의 동작을 용이하게 하는 상이한 플랫폼들을 제공할 수 있다. 예로서, ABC 회사로부터의 메일 서버는 OS 'A'를 요구할 수 있는 반면 XYZ 회사로부터의 메일 서버는 모바일 장치로의/로부터의 데이터를 동기화하기 위해 OS 'B'를 요구할 수 있다. 만일 사용자가 양쪽 ABC 및 XYZ 회

사들의 이메일 서버들과 동기화한다면, 심리스한 동기화를 실행하기 위해 다수의 OS들을 로딩하는 것이 가능할 수 있다. 유사하게, 각 OS는 회사 특정의 이메일 프로토콜들 및 인터페이스들을 이용하기 위해 요구될 수 있다. 여기서, 다중-부팅 기능은 특정한 상황에서 적절한 OS들의 선택을 가능하게 할 수 있다.

[0056] 선택 컴포넌트(602)는 도 5에 관련하여 설명된 부팅 로더에 관련된 기능을 대표할 수 있다. 예를 들면, 선택 컴포넌트(602)는 사용자에게 이용 가능한 OS들의 메뉴를 제공할 수 있다. 동작시에, 사용자는 드롭 다운 메뉴로부터 OS를 선택할 수 있다. 여기에 설명된 본 발명에서 벗어나지 않고 음성 명령, 빠른 버튼, 포인트 앤 클릭(point & click) 메커니즘을 포함하지만 이들에 제한되지 않는 다른 선택 메커니즘들이 이용될 수 있다. 이들 메커니즘들은 특정한 OS를 선택하기 위해, 또는 대안적으로, 원하는 대로 OS 부팅 레벨(예를 들면, 스테이지)을 지정하기 위해 이용될 수 있다.

[0057] 구현 컴포넌트(604)는 OS 또는 그의 스테이지들을 로딩하는 실행 기능을 예시한다. 따라서, 구현 컴포넌트(604)는 본 발명에 따른 OS들의 로딩을 용이하게 하는 코드를 찾아내어 트리거할 수 있다. 예를 들면, 구현 컴포넌트(604)는 OS를 로딩하거나 부팅하기 위해 필요한 파일들 및 데이터를 수용하는 데이터 저장소 내의 적절한 파티션을 찾아낼 수 있다.

[0058] 전술한 바와 같이, 본 발명에서 벗어나지 않고 다른 위치들(예를 들면, 데이터 저장소들) 내에 다수의 OS들이 위치할 수 있다. 이들 다중-위치 또는 다중-저장소 실시예들에서, 구현 컴포넌트(604)는 적절한 위치에 매핑할 수 있고 그 후 OS(또는 OS들의 그룹)의 검색 및 로딩을 용이하게 할 수 있다. 또한, 본 발명의 양태들에 따라 논리적 파티션들이 이용될 수 있다 - 이들 양태들은 이 명세서 및 여기에 첨부된 청구항들의 일부로 간주되어야 한다.

[0059] 이제 도 7을 참조하면, 본 발명의 실시예에 따른 선택 컴포넌트(102)의 예가 도시되어 있다. 일반적으로, 메뉴를 제공하고 안정적인 사용자 명령(예를 들면, 음성 명령)을 제공하는 것 외에도, 선택 컴포넌트(102)는 컨텍스트 기반 선택 컴포넌트(702) 및 아이덴티티 기반 선택 컴포넌트(704)를 포함할 수 있다. 아래 더 상세히 설명되는 바와 같이, 예시된 컴포넌트들(702, 704) 각각과 관련하여 정책 또는 규칙(예를 들면, 구현 방식)이 적용될 수 있다. 또한, 주어진 컨텍스트에서 또는 주어진 아이덴티티에 관련된 것으로서 구현할 적절한 OS(또는 OS의 스테이지)를 결정 및/또는 추론하기 위해 MLR 메커니즘들(또는 로직)이 이용될 수 있다.

[0060] 컨텍스트 기반 컴포넌트(702)는, 환경뿐만 아니라 생리적 센서들(예를 들면, GPS(global position systems), 캘린더, 클록, 걱정 검출기, 마이크, 활동 모니터 등)을 포함할 수 있는 감각 메커니즘들로부터 수신되거나, 액세스되거나, 또는 다른 식으로 얻어지는 정보를 처리함으로써 적절한 OS를 선택하는 판정 프로세스에 컨텍스트 인식을 통합할 수 있다. 바꾸어 말하면, 사용자 환경, 장치 환경, 날짜, 시각, 장소, 약속된 활동, 스케줄/약속, 기업 제휴 등을 포함하지만 이들에 제한되지 않는 컨텍스트 요인들이 설치할 적절한 OS(또는 OS의 스테이지)를 결정(또는 추론)할 때 논리적 판정들에 포함될 수 있다.

[0061] 아이덴티티 기반 선택 컴포넌트(704)는 사용자가 그에 의해 OS가 선택될 수 있는 특정 '아이덴티티' 또는 사람을 정의하는 것을 가능하게 한다. 사용자는 정책들, 선호들, 보안, 데이터 액세스 등에 관련되지만 이들에 제한되지 않는 조건들을 정의할 수 있는 다수의 '아이덴티티들' 또는 사람들을 가질 수 있다. 따라서, 이들 아이덴티티들은 적절한 OS를 결정 및/또는 추론하기 위해 이용될 수 있다.

[0062] 동작시에, 규칙들 또는 구현 방식들은 정책 또는 선호로 정의될 수 있고 이것들은 그 후 선택 조건들을 평가하기 위해 이용될 수 있다. 예시되지는 않았지만, 여기에 설명된 본 발명은 규칙 생성을 가능하게 하는 사용자 인터페이스(UI)를 포함하도록 구상된다는 것을 이해할 것이다. 규칙 기반 로직 외에도, 본 발명은 또한 거의 임의의 수의 요인들에 따라서 OS를 결정 또는 추론하기 위해 MLR 메커니즘들을 이용할 수 있다. 예를 들면, 설치 또는 부팅을 위한 OS를 자동으로 선택하기 위해 아이덴티티(또는 사람)가 이용될 수 있다.

[0063] 규칙 기반의 대안적인 양태들에 따르면, OS를 정의 및/또는 선택하기 위해 구현 방식(예를 들면, 규칙)이 적용될 수 있다. 규칙 기반 구현은 OS의 스테이지를 자동으로 정의하거나 특정된 조건들에 따라 특정한 OS를 선택할 수 있다는 것을 알 것이다.

[0064] 전술한 바와 같이, 사용자는 OS-선택 정책 및/또는 선호를 정의할 수 있는 규칙을 확립하기 위해 UI를 이용할 수 있다. 도 7의 예시적인 양태에서, 선택 컴포넌트(102)는 사용자 정의된 선호 또는 정책에 따라서 프로그램 또는 구성될 수 있다. 또한, 특정 하드웨어 구성에 따라 또는 소프트웨어 애플리케이션에 따라 규칙이 확립될 수 있다. 예를 들면, 특정 메모리 용량 및/또는 장치의 디스플레이에 따라 규칙이 구성될 수 있다. 바꾸어 말하면, 전술한 바와 같이, 하드웨어 장치(예를 들면, 메모리, 프로세서)의 특정 제한들을 고려하도록 규칙이 확

립될 수 있다. 따라서, 일 양태에서, 특정 핸드헬드 장치가 작은 메모리 용량을 가진다면, 덜 강건한 OS를 선택하거나 또는 대안적으로, 완전한 OS를 로딩하기보다는 OS의 초기 스테이지를 선택하도록 규칙이 생성될 수 있다.

[0065] 규칙 기반 로직 외에도, 본 발명은 본 발명에 따른 하나 이상의 특징의 자동화를 용이하게 하는 MLR 로직을 이용할 수 있다. 본 발명은 (예를 들면, OS 선택과 관련하여) 그의 다양한 양태들을 수행하기 위해 다양한 MLR 기반 방식들을 이용할 수 있다. 예를 들면, 주어진 컨텍스트에서 또는 특정한 아이덴티에 관련하여 어느 OS를 로딩할지를 결정하기 위한 프로세스가 자동 분류자(classifier) 시스템 및 프로세스를 통해 용이해질 수 있다.

[0066] 분류자는 입력 속성 벡터 $x = (x_1, x_2, x_3, x_4, x_n)$ 을 그 입력이 클래스에 속하는 신뢰(confidence)에 매핑하는 함수로서, 즉, $f(x) = confidence(class)$ 이다. 그러한 분류는 사용자가 자동으로 수행되기를 원하는 액션을 예지하거나 추천하기 위해 확률 및/또는 통계 기반 분석(예를 들면, 분석 유틸리티들 및 비용들을 계산에 넣는 것)을 이용할 수 있다.

[0067] SVM(support vector machine)은 이용될 수 있는 분류자의 예이다. SVM은 가능한 입력들의 공간에서 초곡면(hypersurface)을 찾아내는 것에 의해 동작하고, 그 초곡면은 비트리거링(non-triggering) 이벤트들로부터 트리거링 기준들(triggering criteria)을 분리시키려 한다. 직관적으로, 이것은 트레이닝 데이터에 가깝지만 동일하지는 않은 테스트 데이터에 대하여 분류를 정확하게 만든다. 다른 방향성(directed) 및 비방향성(undirected) 모델 분류 접근법들은, 예를 들면, 나이브 베이즈(naive Bayes), 베이즈 네트워크, 관정 트리, 신경 네트워크, 퍼지 로직 모델을 포함하고, 상이한 독립 패턴들을 제공하는 확률 분류 모델들이 이용될 수 있다. 여기서 사용되는 분류는 또한 우선 순위의 모델들을 개발하기 위해 이용되는 통계적 회귀(statistical regression)를 포함한다.

[0068] 본 명세서로부터 쉽게 이해되는 바와 같이, 본 발명은 (예를 들면, 일반적인 트레이닝 데이터를 통해) 명시적으로 트레이닝될 뿐만 아니라 (예를 들면, 사용자 행동을 관찰하고, 외래의 정보를 수신하는 것을 통해) 암시적으로 트레이닝되는 분류자들을 이용할 수 있다. 예를 들면, SVM들은 분류자 생성자(classifier constructor) 및 특징 선택 모듈 내의 학습 또는 트레이닝 국면을 통해 구성된다. 따라서, 분류자(들)는 미리 결정된 조건들에 따라 어느 OS를 선택할지 또는 애플리케이션, 컨텍스트, 아이덴티티 등을 고려하여 OS의 어느 부분(예를 들면, 스테이지, 레벨)을 로딩할지를 결정하는 것을 포함하지만 이에 제한되지 않는 다수의 기능들을 자동으로 학습하고 수행하기 위해 이용될 수 있다.

[0069] 도 8은 파티션 로케이터 컴포넌트(partition locator component)(802)를 포함하는 구현 컴포넌트(104)의 예의 블록도를 예시한다. 이 서브-컴포넌트(802)는 주어진 상황에서 또는 특정 선택에 따라서 특정한 OS의 부팅을 실행하기 위하여 특정 위치를 찾아내고 특정 위치에 매핑할 수 있다. 전술한 바와 같이, OS들은 단일 디스크의 다른 파티션들 내에 위치할 수 있고 또는, 대안적으로, 모바일 장치에 의해 액세스되는 상이한 적합한 위치들에 위치할 수 있다.

[0070] 이제 도 9를 참조하면, 본 발명의 일 양태에 따른 휴대용 장치(900)의 개략 블록도가 예시되어 있고, 여기서 프로세서(902)는 장치(900)의 일반적인 동작을 제어를 담당한다. 휴대용 장치(900)는 휴대폰, 스마트폰, PDA, 개인용 음악 플레이어, 이미지 캡처 장치(예를 들면, 카메라), 개인용 게임 스테이션 등을 포함하지만 이들에 제한되지 않는 거의 임의의 휴대용 장치를 대표할 수 있다는 것을 이해해야 한다.

[0071] 프로세서(902)는 여기에 설명된 다양한 기능들을 수행하기 위해 장치(900) 내의 다양한 컴포넌트들을 제어하고 동작시키도록 프로그램될 수 있다. 프로세서(902)는 복수의 적합한 프로세서들 중 임의의 것일 수 있다. 프로세서(902)가 본 발명에 관한 기능들을 수행하도록 프로그램될 수 있는 방식은 여기에 제공된 설명에 기초하여 당해 기술의 통상의 기술을 가진 당업자들에게 명백할 것이다.

[0072] 프로세서(902)에 접속된 메모리 및 저장 컴포넌트(904)는 프로세서(902)에 의해 실행되는 프로그램 코드를 저장하도록 기능하고, 또한 데이터, 서비스, 메타데이터, 장치 상태 등과 같은 정보를 저장하기 위한 저장 수단으로서 기능한다. 또한, 다른 양태들에서, 메모리 및 저장 컴포넌트(904)는 독립 실행형 저장 장치일 수 있고 또는 클라우드 또는 다른 네트워크 기반 저장 수단(예를 들면, 익스체인지(Exchange) 서버)와 동기화됨으로써 로컬 온-보드 저장을 확립할 수 있다.

[0073] 메모리(904)는 적어도 획득되는 정보의 완전한 세트를 저장하도록 적합하게 적응된 비휘발성 메모리일 수 있다. 따라서, 메모리(904)는 프로세서(902)에 의한 고속 액세스를 위한 RAM 또는 플래시 메모리 및/또는 텍스트, 이미지, 오디오, 및 비디오 콘텐츠를 포함하는 수 기가바이트의 데이터를 저장할 수 있는 대용량 저장 메모리, 예

를 들면, 마이크로 드라이브를 포함할 수 있다. 일 양태에 따르면, 메모리(904)는 다른 서비스들/OS들에 관한 다수의 정보 세트들을 저장하기에 충분한 저장 용량을 갖고, 프로세서(902)는 다른 서비스들/OS들에 대응하는 다양한 정보 세트들 사이에 교대하거나 순환하기 위한 프로그램을 포함할 수 있다.

[0074] 디스플레이(906)는 디스플레이 드라이버 시스템(908)을 통해 프로세서(902)에 연결될 수 있다. 디스플레이(906)는 컬러 LCD(liquid crystal display), 플라즈마 디스플레이, 터치 스크린 디스플레이 등일 수 있다. 일례로, 디스플레이(906)는 터치 스크린 디스플레이이다. 디스플레이(906)는 데이터, 그래픽, 또는 다른 정보 콘텐츠를 제공하도록 기능한다. 또한, 디스플레이(906)는 장치(900)의 실행을 제어하는 각종 기능들을 디스플레이할 수 있다. 예를 들면, 터치 스크린 예에서, 디스플레이(906)는 사용자가 장치(900)의 기능들과 보다 쉽게 인터페이스하는 것을 용이하게 할 수 있는 터치 선택 버튼들을 디스플레이할 수 있다.

[0075] 온보드 전원 시스템(910)(예를 들면, 배터리 팩)에 의해 프로세서(902) 및 장치(900)를 형성하는 다른 컴포넌트들에 전력이 제공될 수 있다. 전원 시스템(910)이 고장나거나 장치(900)로부터 분리되는 경우에는, 프로세서(902)(및 다른 컴포넌트들(예를 들면, 센서, 이미지 캡처 장치))에 전력을 제공하고 온보드 전원 시스템(910)을 충전하기 위해 보조 전원(912)이 이용될 수 있다. 장치(900)의 프로세서(902)는 예상되는 전원 고장의 검출시에 전류 소비를 감소시키기 위해 절전 모드(sleep mode)를 유도할 수 있다.

[0076] 장치(900)는 프로세서(902)를 원격 컴퓨터, 서버, 서비스 등과 인터페이스하기 위해 이용되는 데이터 통신 포트(916)를 갖는 통신 서비스시스템(914)을 포함한다. 포트(916)는 USB(Universal Serial Bus) 및 IEEE 1394 시리얼 통신 능력 중 적어도 하나를 포함할 수 있다. 다른 기술들은 또한, 예를 들면, 적외선 데이터를 이용하는 적외선 통신, 블루투스(Bluetooth™) 등을 포함할 수 있지만, 이들에 제한되는 것은 아니다.

[0077] 장치(900)는 또한 프로세서(902)와 실효적으로 통신하는 무선 주파수(RF) 트랜스미버 섹션(918)을 포함할 수 있다. RF 섹션(918)은 RF 수신기(920)를 포함하고, 이는 안테나(922)를 통해 원격 장치로부터 RF 신호들을 수신하고 그 신호에 변조된 디지털 정보를 얻기 위해 그 신호를 복조할 수 있다. RF 섹션(918)은 또한, 예를 들면, 사용자 입력(926)(예를 들면, 키패드)을 통한 수동 사용자 입력에 응답하여 또는 통신 범위 또는 다른 미리 정해진 및 프로그램된 기준들에 들어가는 것의 검출 및/또는 그것에서 나가는 것의 예상에 응답하여 자동으로 원격 장치에 정보(예를 들면, 데이터, 서비스)를 송신하기 위한 RF 송신기(924)를 포함한다.

[0078] 선택 컴포넌트(928)는 정책, 선호, 컨텍스트 등에 따라 적절한 OS 또는 OS의 스테이지를 선택하기 위해 이용될 수 있다. 따라서, 구현 컴포넌트(930)는 선택에 따라 OS를 찾아내어 로딩하기 위해 이용될 수 있다. 또한, 본 발명의 하나 이상의 특징들을 자동화하기 위해 옵션의 MLR 컴포넌트(932)가 이용될 수 있다. 위에 더 상세히 설명된 바와 같이, 자동화 액션을 실행하기 위해 MLR 컴포넌트(932)(및/또는 규칙 기반 로직 컴포넌트(도시되지 않음))가 이용될 수 있다. 이들 컴포넌트들은 위에 설명된 유사한 컴포넌트들(및 서브-컴포넌트들)의 기능을 가능하게 할 수 있다는 것을 알아야 한다.

[0079] 이제 도 10을 참조하면, 개시된 아키텍처를 실행하도록 동작 가능한 컴퓨터의 블록도가 예시되어 있다. 본 발명의 다양한 양태들에 대한 추가적인 컨텍스트를 제공하기 위하여, 도 10 및 다음의 설명은 본 발명의 다양한 양태들이 구현될 수 있는 적합한 컴퓨팅 환경(1000)의 간략하고 일반적인 설명을 제공하기 위한 것이다. 본 발명은 위에서 하나 이상의 컴퓨터에서 실행될 수 있는 컴퓨터 실행가능한 명령어들의 일반적인 컨텍스트에서 설명되었지만, 당업자들은 본 발명은 또한 다른 프로그램 모듈들과 조합하여 및/또는 하드웨어와 소프트웨어의 조합으로서 구현될 수 있다는 것을 인지할 것이다.

[0080] 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 컴포넌트, 데이터 구조 등을 포함한다. 또한, 당업자들은 본 발명의 방법들은 단일 프로세서 또는 멀티프로세서 컴퓨터 시스템, 미니컴퓨터, 메인프레임 컴퓨터뿐만 아니라, 퍼스널 컴퓨터, 핸드헬드 컴퓨팅 장치, 마이크로프로세서 기반 또는 프로그램가능한 가전 제품 등 - 그 각각은 하나 이상의 관련된 장치들에 실효적으로 연결될 수 있음 - 을 포함하는 다른 컴퓨터 시스템 구성에 의해 실시될 수 있다는 것을 알 것이다.

[0081] 본 발명의 예시된 양태들은 또한 통신 네트워크를 통해 링크된 원격 처리 장치들에 의해 특정 태스크가 수행되는 분산 컴퓨팅 환경들에서 실행될 수도 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 로컬 및 원격 메모리 저장 장치 둘다에 위치할 수 있다.

[0082] 컴퓨터는 통상적으로 각종 컴퓨터 판독가능 매체를 포함한다. 컴퓨터에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 이동식 및 이동불가식 매체를 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수

있지만 이에 제한되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위해 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 이동불가식 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광 디스크 저장 장치, 자기 카세트, 자기 테이프, 자기 디스크 저장 장치 또는 기타 자기 저장 장치, 또는 컴퓨터에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 이용될 수 있는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다.

[0083] 통신 매체는 통상적으로 반송파(carrier wave) 또는 기타 전송 메커니즘(transport mechanism)과 같은 피변조 데이터 신호(modulated data signal)에 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터 등을 구현하고 임의의 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는, 신호에 정보를 인코딩하도록 그 신호의 하나 이상의 특성을 설정 또는 변경시킨 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 접속과 같은 유선 매체, 그리고 음향, RF, 적외선 및 기타 무선 매체와 같은 무선 매체를 포함한다. 상기한 것들 중 임의의 것들의 조합이 또한 컴퓨터 판독가능 매체의 영역 안에 포함되어야 한다.

[0084] 도 10을 다시 참조하면, 본 발명의 다양한 양태들을 구현하기 위한 예시적인 환경(1000)은 컴퓨터(1002)를 포함하고, 컴퓨터(1002)는 처리 장치(1004), 시스템 메모리(1006), 및 시스템 버스(1008)를 포함한다. 시스템 버스(1008)는 시스템 메모리(1006)를 포함하지만 이에 제한되지 않는 컴포넌트들을 처리 장치(1004)에 연결한다. 처리 장치(1004)는 다양한 이용 가능한 프로세서들 중 어떤 것이라도 될 수 있다. 듀얼 마이크로프로세서 및 기타 멀티프로세서 아키텍처도 처리 장치(1004)로서 채용될 수 있다.

[0085] 시스템 버스(1008)는 (메모리 컨트롤러를 갖는 또는 메모리 컨트롤러가 없는) 메모리 버스, 주변 버스, 및 각종의 상업적으로 입수 가능한 버스 아키텍처들 중 임의의 것을 이용하는 로컬 버스에 더 상호접속될 수 있는 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 시스템 메모리(1006)는 ROM(read-only memory)(1010) 및 RAM(random access memory)(1012)를 포함한다. 기본 입력/출력 시스템(BIOS)이 ROM, EPROM, EEPROM과 같은 비휘발성 메모리(1010)에 저장되고, BIOS는 시동 중과 같이 컴퓨터(1002) 내의 엘리먼트들 사이에 정보를 전송하는 기본 루틴들을 포함한다. RAM(1012)은 또한 데이터를 캐싱하기 위한 스택 RAM과 같은 고속 RAM을 포함할 수 있다.

[0086] 컴퓨터(1002)는 내부 하드 디스크 드라이브(HDD)(1014)(예를 들면, EIDE, SATA) - 이 내부 하드 디스크 드라이브(1014)는 또한 적합한 새시(도시되지 않음)에서 외부 사용을 위해 구성될 수도 있음 -, 자기 플로피 디스크 드라이브(FDD)(1016)(예를 들면, 이동식 디스켓(1018)으로부터 판독하거나 그것에 기입하기 위한 것) 및 광학 디스크 드라이브(1020)(예를 들면, CD-ROM 디스크(1022)를 판독하거나, DVD와 같은 다른 고용량 광학 매체로부터 판독하거나 그것에 기입하기 위한 것)를 더 포함한다. 하드 디스크 드라이브(1014), 자기 디스크 드라이브(1016) 및 광학 디스크 드라이브(1020)는 각각 하드 디스크 드라이브 인터페이스(1024), 자기 디스크 드라이브 인터페이스(1026) 및 광학 드라이브 인터페이스(1028)에 의해 시스템 버스(1008)에 접속될 수 있다. 외부 드라이브 구현을 위한 인터페이스(1024)는 USB(Universal Serial Bus) 및 IEEE 1394 인터페이스 기술들 중 적어도 하나 또는 양쪽 모두를 포함한다. 다른 외부 드라이브 접속 기술들이 본 발명의 구상 안에 있다.

[0087] 드라이브들 및 이들과 관련된 컴퓨터 저장 매체는, 데이터, 데이터 구조, 컴퓨터 판독가능 명령어 등의 비휘발성 저장을 제공한다. 컴퓨터(1002)에 대하여, 드라이브들 및 매체는 적합한 디지털 포맷의 임의의 데이터의 저장을 수용한다. 비록 상기 컴퓨터 판독가능 매체의 설명은 HDD, 이동식 자기 디스켓, 및 CD 또는 DVD와 같은 이동식 광학 매체를 참조하지만, 당업자들은 집 드라이브, 자기 카세트, 플래시 메모리 카드, 카트리지 등과 같은, 컴퓨터에 의해 판독가능한 다른 유형의 매체들도 예시적인 동작 환경에서 이용될 수 있고, 또한 임의의 그러한 매체들은 본 발명의 방법들을 수행하기 위한 컴퓨터 실행가능한 명령어들을 포함할 수 있다는 것을 알 것이다.

[0088] 드라이브들 및 RAM(1012)에는, 운영 체제(1030), 하나 이상의 애플리케이션 프로그램(1032), 기타 프로그램 모듈(1034) 및 프로그램 데이터(1036)를 포함하는 다수의 프로그램 모듈이 저장될 수 있다. 운영 체제, 애플리케이션, 모듈, 및/또는 데이터의 전부 또는 일부는 또한 RAM(1012)에 캐싱될 수도 있다. 본 발명은 다양한 상업적으로 입수 가능한 운영 체제들 또는 운영 체제들의 조합들과 함께 구현될 수 있다는 것을 알아야 한다.

[0089] 사용자는 하나 이상의 유선/무선 입력 장치, 예를 들면, 키보드(1038) 및 마우스(1040)와 같은 포인팅 장치를 통해 명령 및 정보를 컴퓨터(1002)에 입력할 수 있다. 다른 입력 장치(도시되지 않음)로는 마이크, IR 리모트 컨트롤, 조이스틱, 게임 패드, 스타일러스 펜, 터치 스크린 등을 포함할 수 있다. 이들 및 기타 입력 장치는 종종 시스템 버스(1008)에 연결된 사용자 입력 인터페이스(1042)를 통해 처리 장치(1004)에 접속되지만, 병

렬 포트, IEEE 1394 직렬 포트, 게임 포트, USB 포트, IR 인터페이스 등과 같은 다른 인터페이스들에 의해 접속될 수도 있다.

[0090] 모니터(1044) 또는 다른 유형의 디스플레이 장치도 비디오 어댑터(1046) 등의 인터페이스를 통해 시스템 버스(1008)에 접속될 수 있다. 모니터(1044) 외에도, 컴퓨터는 통상적으로 스피커, 프린터 등의 기타 주변 출력 장치(도시되지 않음)를 포함한다.

[0091] 컴퓨터(1002)는 원격 컴퓨터(들)(1048)와 같은 하나 이상의 원격 컴퓨터로의 유선 및/또는 유선 통신을 통한 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(들)(1048)는 워크스테이션, 서버 컴퓨터, 라우터, 퍼스널 컴퓨터, 휴대용 컴퓨터, 마이크로프로세서 기반 엔터테인먼트 전자 기기, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 통상적으로 컴퓨터(1002)에 관련하여 설명된 구성요소들의 다수 또는 그 전부를 포함하지만, 간결함을 위하여, 메모리/저장 장치(1050)만 예시되어 있다. 도시된 논리적 접속들은 LAN(local area network)(1052) 및/또는 보다 큰 네트워크, 예를 들면, WAN(wide area network)(1054)으로의 유선/무선 접속을 포함한다. 그러한 LAN 및 WAN 네트워킹 환경은 사무실 및 회사에서 흔한 것이고, 인터넷들과 같은 전사적(enterprise-wide) 컴퓨터 네트워크들을 용이하게 하고, 이들 모두는 글로벌 통신 네트워크, 예를 들면, 인터넷에 접속할 수 있다.

[0092] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(1002)는 유선 및/또는 무선 네트워크 인터페이스 또는 어댑터(1056)를 통해 로컬 네트워크(1052)에 접속된다. 어댑터(1056)는 LAN(1052)으로의 유선 또는 무선 통신을 용이하게 할 수 있고, LAN(1052)도 무선 어댑터(1056)와 통신하기 위해 그 위에 배치된 무선 액세스 포인트를 포함할 수 있다.

[0093] WAN 네트워킹 환경에서 사용될 때, 컴퓨터(1002)는 모뎀(1058)을 포함할 수 있고, 또는 WAN(1054) 상의 통신 서버에 접속되고, 또는 WAN(1054)을 통해, 예를 들면 인터넷을 통해 통신을 설정하기 위한 다른 수단을 갖는다. 내장형 또는 외장형이고 무선 또는 유선 장치인 모뎀(1058)은 시리얼 포트 인터페이스(1042)를 통해 시스템 버스(1008)에 접속된다. 네트워크화된 환경에서, 컴퓨터(1002) 또는 그의 일부와 관련하여 도시된 프로그램 모듈은 원격 메모리/저장 장치(1050)에 저장될 수 있다. 도시된 네트워크 접속은 예시적인 것이며 컴퓨터들 사이의 통신 링크를 설정하는 다른 수단이 사용될 수 있다는 것을 이해할 것이다.

[0094] 컴퓨터(1002)는 무선 통신에서 실효적으로 배치된 임의의 무선 장치들 또는 엔티티들, 예를 들면, 프린터, 스캐너, 데스크톱 및/또는 휴대용 컴퓨터, 휴대용 정보 단말기, 통신 위성, 무선으로 검출가능한 태그와 관련된 임의의 장비 또는 장소(예를 들면, 키오스크, 가판대, 화장실), 및 전화기와 통신하도록 동작 가능하다. 이것은 적어도 Wi-Fi 및 블루투스 무선 기술들을 포함한다. 따라서, 통신은 통상의 네트워크에서와 같이 미리 정의된 구조이거나 또는 단순히 적어도 2개의 장치들 사이의 애드 혹(ad hoc) 통신일 수 있다.

[0095] Wi-Fi(즉, Wireless Fidelity)는 집의 소파, 호텔방의 침대, 또는 직장의 회의실로부터 선 없이 인터넷에 접속하는 것을 가능하게 한다. Wi-Fi는 그러한 장치, 예를 들면, 컴퓨터가 실내 및 외에서, 기지국의 범위 내의 어디에서든, 데이터를 송수신할 수 있게 하는 휴대폰에서 이용되는 것과 유사한 무선 기술이다. Wi-Fi 네트워크들은 안전하고, 신뢰할 수 있고, 빠른 무선 접속을 제공하기 위해 IEEE 802.11(a,b,g 등)이라고 불리는 무선 기술들을 이용한다. Wi-Fi 네트워크는 컴퓨터들을 서로에 대해, 인터넷에, 및 (IEEE 802.3 또는 이더넷을 이용하는) 유선 네트워크에 접속하기 위해 이용될 수 있다. Wi-Fi 네트워크들은 비허가 2.4 및 5 GHz 무선 대역들에서, 예를 들면, 11 Mbps(802.11a) 또는 54 Mbps(802.11b) 데이터 레이트에서, 또는 양쪽 대역들(듀얼 밴드)을 포함하는 제품들과 함께 동작하고, 따라서 그 네트워크들은 많은 사무실들에서 사용되는 베이직 10BaseT 유선 이더넷 네트워크와 유사한 실세계 성능을 제공할 수 있다.

[0096] 이제 도 11을 참조하면, 본 발명에 따른 예시적인 컴퓨팅 환경(1100)의 개략 블록도가 예시되어 있다. 시스템(1100)은 하나 이상의 클라이언트(들)(1102)를 포함한다. 클라이언트(들)(1102)는 하드웨어 및/또는 소프트웨어(예를 들면, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 클라이언트(들)(1102)는, 예를 들면, 본 발명을 이용함으로써, 쿠키(들) 및/또는 관련된 컨텍스트 정보를 수용할 수 있다.

[0097] 시스템(1100)은 또한 하나 이상의 서버(들)(1104)를 포함한다. 서버(들)(1104)도 하드웨어 및/또는 소프트웨어(예를 들면, 스레드, 프로세스, 컴퓨팅 장치)일 수 있다. 서버(들)(1104)은, 예를 들면, 본 발명을 이용함으로써 변환들을 수행하는 스레드들을 수용할 수 있다. 클라이언트(1102)와 서버(1104) 사이의 하나의 가능한 통신은 2개 이상의 컴퓨터 프로세스들 사이에 송신되도록 적용된 데이터 패킷들의 형태를 가질 수 있다. 데이터 패킷은, 예를 들면, 쿠키들 및/또는 관련된 컨텍스트 정보를 포함할 수 있다. 시스템(1100)은 클라이언트

(들)(1102)와 서버(들)(1104) 사이의 통신을 용이하게 하기 위해 이용될 수 있는 통신 프레임워크(1106)(예를 들면, 인터넷과 같은 글로벌 통신 네트워크)를 포함한다.

[0098]

통신은 유선(광섬유를 포함함) 및/또는 무선 기술을 통해 용이하게 될 수 있다. 클라이언트(들)(1102)는 클라이언트(들)(1102)에 로컬인 정보(예를 들면, 쿠키(들) 및/또는 관련된 컨텍스트 정보)를 저장하기 위해 이용될 수 있는 하나 이상의 클라이언트 데이터 저장소(들)(1108)에 실제로 접속된다. 유사하게, 서버(들)(1104)는 서버(들)(1104)에 로컬인 정보를 저장하기 위해 이용될 수 있는 하나 이상의 서버 데이터 저장소(들)(1110)에 실제로 접속된다.

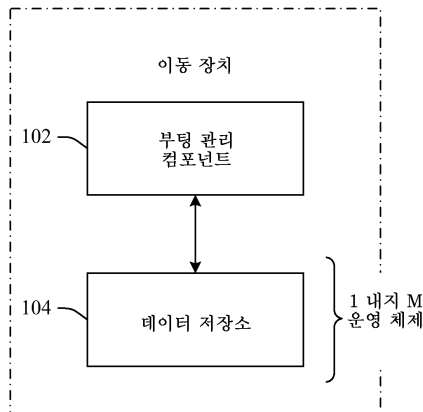
[0099]

위에 설명된 것은 본 발명의 예들을 포함한다. 물론, 본 발명을 설명하기 위해 생각할 수 있는 컴포넌트들 또는 방법들의 모든 조합을 다 설명하는 것은 불가능하지만, 통상의 기술을 가진 당업자라면 본 발명의 다수의 추가적인 조합들 및 치환들이 가능하다는 것을 알 수 있다. 따라서, 본 발명은 첨부된 청구항들의 정신 및 범위 내에 있는 모든 그러한 변경들, 수정들 및 변형들을 망라하도록 되어 있다. 또한, 용어 "includes"가 상세한 설명 또는 청구항에서 사용되는 한은, 그러한 용어는 "comprising"이 청구항에서 전환 단어(transitional word)로서 이용될 때 해석되듯이, "comprising"과 유사한 방식으로 포괄적(inclusive)이도록 되어 있다.

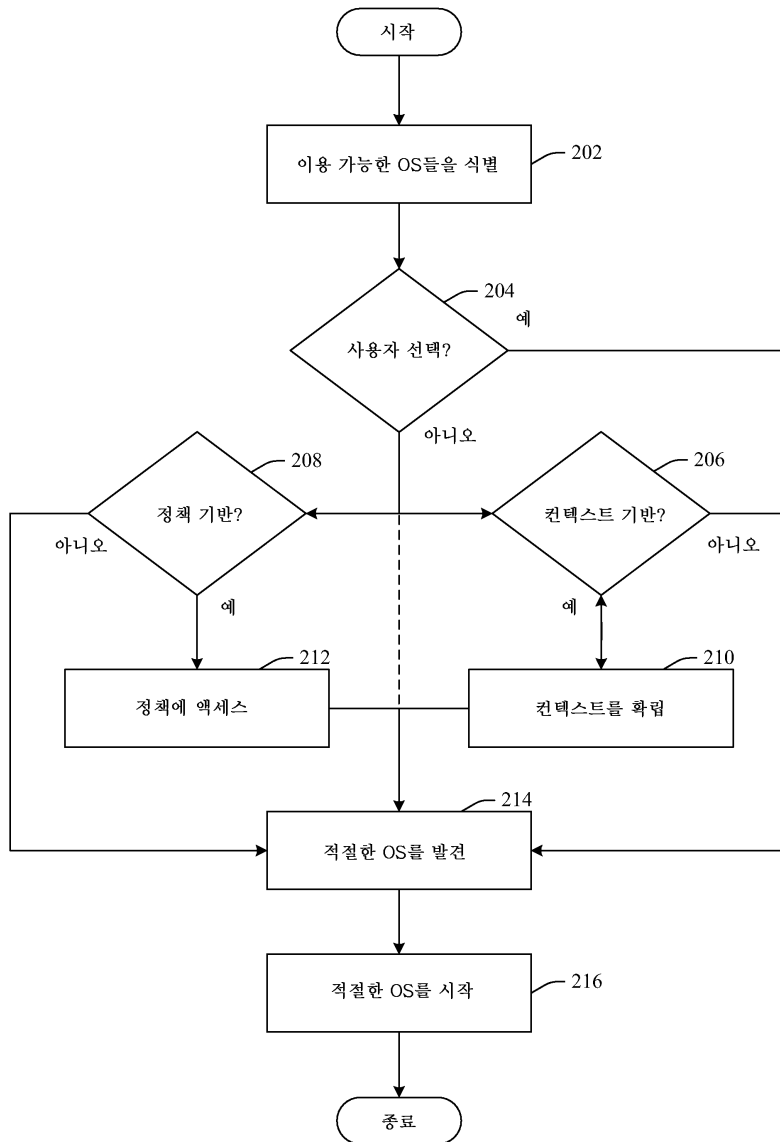
도면

도면1

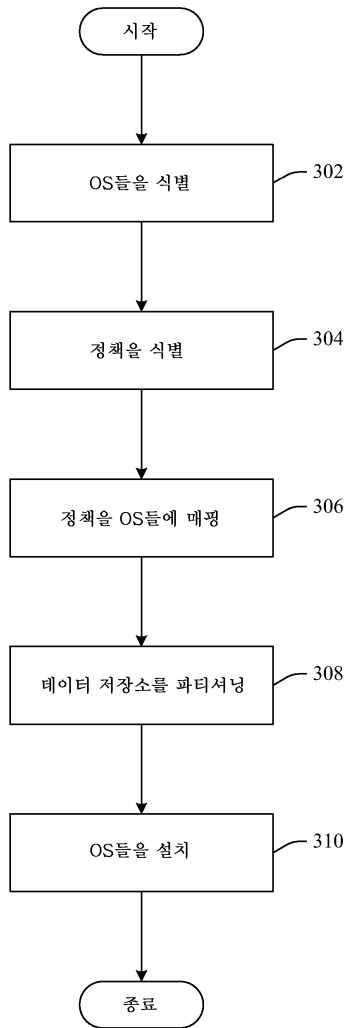
↖ 100



도면2

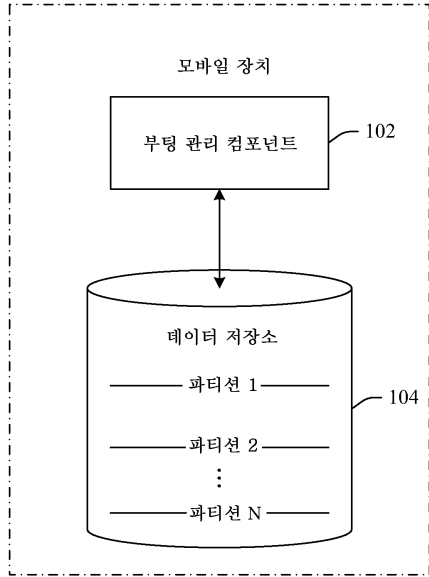


도면3

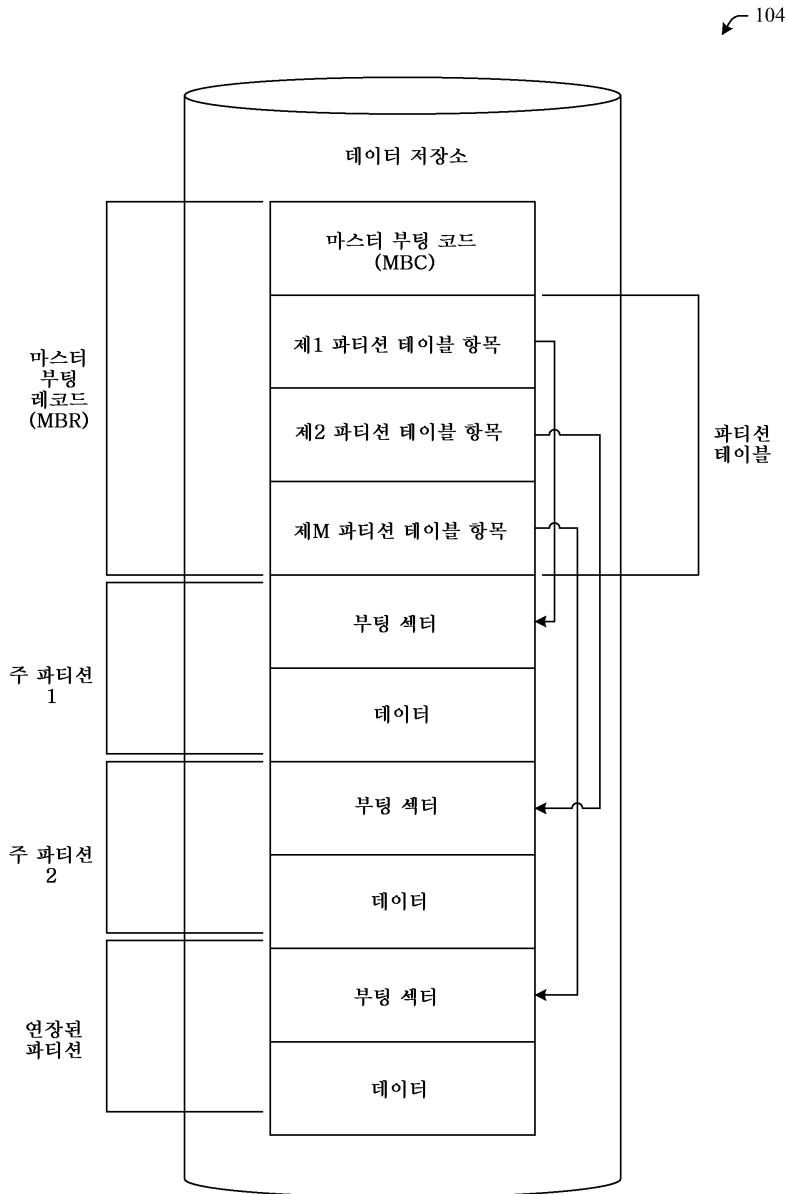


도면4

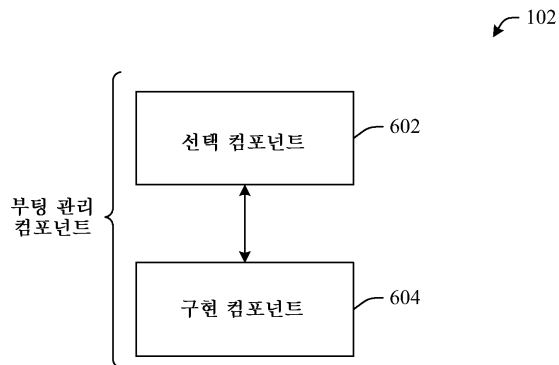
100



도면5

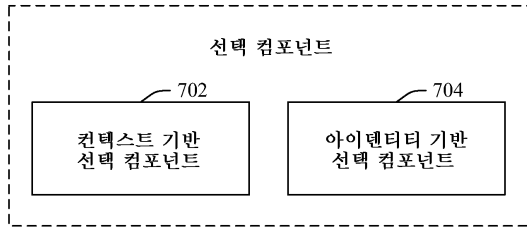


도면6



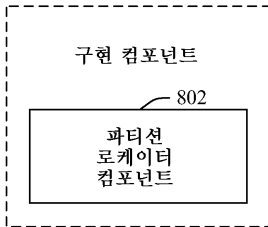
도면7

102



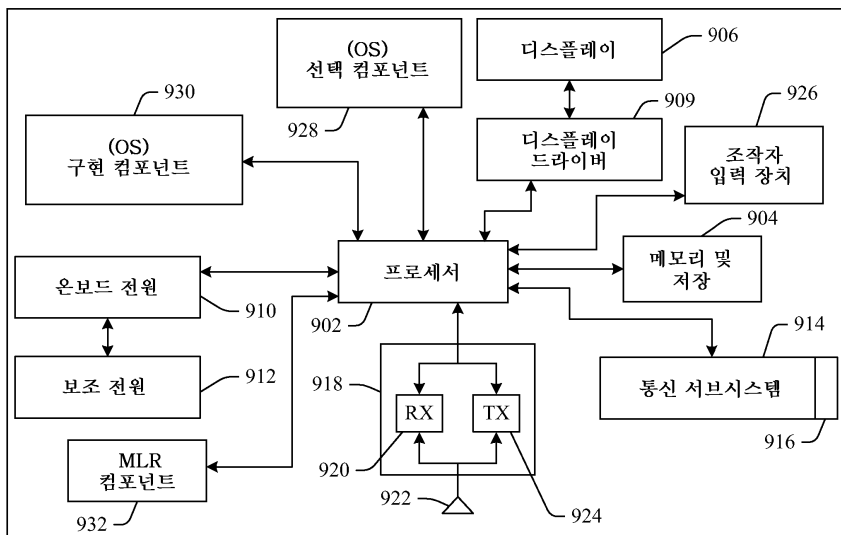
도면8

104

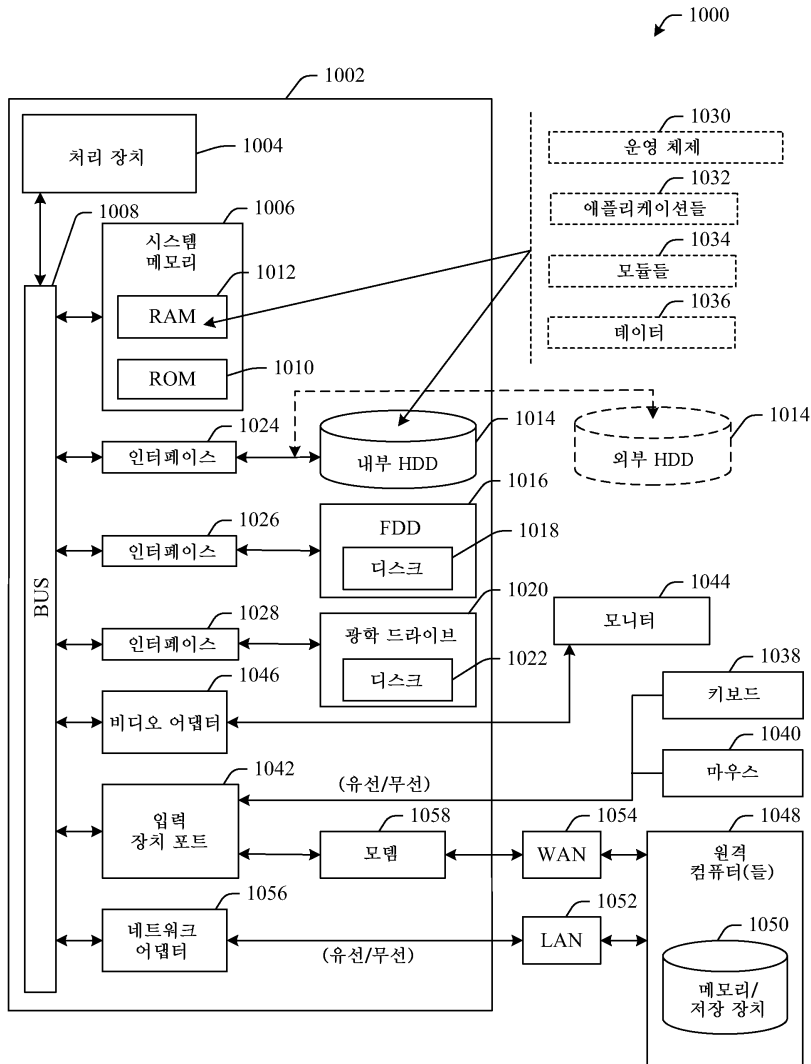


도면9

900



도면10



도면11

