(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년02월03일
(11) 등록번호 10-1702700
(24) 등록일자 2017년01월26일

(51) 국제특허분류(Int. Cl.)
G06F 9/44 (2006.01) *G06F 1/32* (2006.01)
(21) 출원번호 10-2013-7014522
(22) 출원일자(국제) 2010년12월18일
심사청구일자 2015년11월18일
(85) 번역문제출일자 2013년06월05일
(65) 공개번호 10-2013-0127465
(43) 공개일자 2013년11월22일
(86) 국제출원번호 PCT/US2010/061180
(87) 국제공개번호 WO 2012/078175
국제공개일자 2012년06월14일
(30) 우선권주장
12/960,835 2010년12월06일 미국(US)
(56) 선행기술조사문헌
US5696897 A*
US6115824 A*
US20010039612 A1*
US20060242398 A1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
마이크로소프트 테크놀로지 라이센싱, 엘엘씨
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
아이건 메메트
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
백 에브게니
미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션
(뒷면에 계속)
(74) 대리인
제일특허법인

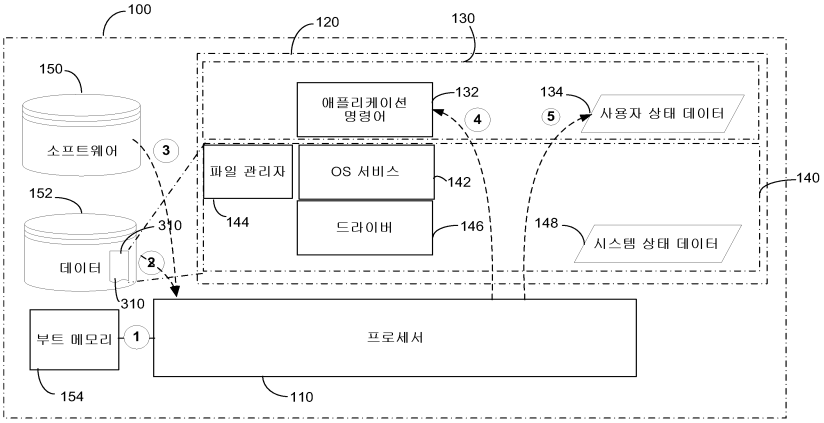
전체 청구항 수 : 총 20 항 심사관 : 김경완

(54) 발명의 명칭 **고속 컴퓨터 시동**

(57) 요 약

셧다운 명령어의 수신이 있을 시, 목표 상태를 나타내는 상태 정보를 기록함으로써, 고속 컴퓨터 시동이 제공된다. 이 목표 상태에서, 컴퓨팅 장치가 모든 사용자 세션을 닫아서, 어떠한 사용자 상태 정보도 목표 상태에 포함되지 않을 수 있다. 그러나 운영 체제는 여전히 실행 중일 수 있다. 컴퓨터를 시동하라는 명령어에 응답하여, 이 목표 상태가 기록된 목표 상태 정보로부터 빠르게 재확립될 수 있다. 시동 시퀀스의 일부분은 사용자 상태를 확립하는 것을 포함한 시동 프로세스를 완료하도록 수행될 수 있다. 셧다운 명령어에 응답하는 변화에도 불구하고 사용자 기대를 보존하기 위해, 기록된 상태 정보를 유지하는 파일의 생성 및 사용이 동적으로 결정된 이벤트에 따라 조건부로 이뤄질 수 있다. 또한 사용자 및 프로그래밍 인터페이스가 기록된 상태 정보의 생성 또는 사용을 무효화하기 위한 옵션을 제공할 수 있다.

대 표 도



(72) 발명자

윌슨 에밀리 엔

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

스타크 커스텐 브이

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

장 수슈

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

스테멘 패트릭 엘

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

킹 브라이언 이

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

카라고우니스 바실리오스

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

제인 닐

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 어텐션: 엘씨에이 - 인터내셔널 페이턴츠 마이크로소프트 코포레이션

명 세 서

청구범위

청구항 1

복수의 하드웨어 구성요소를 포함하는 컴퓨팅 장치 상에서 수행되는 방법으로서,

상기 복수의 하드웨어 구성요소는 상기 컴퓨팅 장치의 임의의 사용자 상태와 상기 컴퓨팅 장치의 시스템 상태를 유지하는 휘발성 메모리 및 프로세서를 포함하고,

상기 방법은 상기 컴퓨팅 장치의 상기 프로세서에 의해 수행되며, 셧다운 명령어(shutdown command)를 수신한 것에 응답하여 상기 컴퓨팅 장치를 셧다운시키도록 구성된 셧다운 시퀀스를 수행하는 단계를 포함하되,

상기 셧다운 시퀀스는,

상기 컴퓨팅 장치 상에서 임의의 사용자 세션을 해제(break down)하는 동작 - 상기 해제에 의해 상기 사용자 상태는 더 이상 상기 휘발성 메모리에 유지되지 않음 - 과,

상기 해제하는 동작에 후속하여, 상기 휘발성 메모리의 콘텐츠를 비휘발성 메모리에 복사하는 동작 - 상기 복사된 콘텐츠는 상기 유지된 시스템 상태는 포함하지만 더 이상 유지되지 않는 상기 사용자 상태는 포함하지 않음 - 과,

상기 컴퓨팅 장치의 상기 복수의 하드웨어 구성요소의 적어도 일부분을 파워 다운(power down)시키는 동작을 포함하는

방법.

청구항 2

제 1 항에 있어서,

상기 해제하는 동작은 상기 시스템 상태로부터 사용자 특정 설정을 제거하는 동작을 포함하는

방법.

청구항 3

제 1 항에 있어서,

상기 셧다운 시퀀스는 상기 파워 다운시키는 동작에 후속하여 또한 시동(startup) 명령어를 수신한 것에 응답하여, 상기 복수의 하드웨어 구성요소 중 적어도 상기 일부분을 파워 업시키는 동작을 더 포함하는

방법.

청구항 4

제 3 항에 있어서,

상기 셧다운 시퀀스는 상기 파워 업시키는 동작에 후속하여, 상기 복사된 콘텐츠를 상기 비휘발성 메모리로부터 상기 휘발성 메모리로 복사하는 동작을 더 포함하는

방법.

청구항 5

제 1 항에 있어서,

상기 복사되는 콘텐츠를 복사하는 동작은 상기 복사되는 콘텐츠를 포함하는 파일을 검출하는 것에 기초하는

방법.

청구항 6

제 1 항에 있어서,

상기 콘텐츠를 복사하는 동작은 파일을 생성하는 동작을 포함하는

방법.

청구항 7

제 1 항에 있어서,

상기 셧다운 시퀀스는 더티 데이터(dirty data)를 상기 비휘발성 메모리에 플러싱(flush)하는 동작을 더 포함하는

방법.

청구항 8

복수의 하드웨어 구성요소를 포함하는 컴퓨팅 장치에 의해 수행되는 경우, 상기 컴퓨팅 장치로 하여금 셧다운 명령어를 수신한 것에 응답하여 상기 컴퓨팅 장치를 셧다운시키도록 구성된 셧다운 시퀀스를 수행하게 하는 컴퓨터 실행가능 명령어들을 저장하는 적어도 하나의 컴퓨터 저장 장치로서,

상기 복수의 하드웨어 구성요소는 상기 컴퓨팅 장치의 임의의 사용자 상태와 상기 컴퓨팅 장치의 시스템 상태를 유지하는 휘발성 메모리를 포함하고,

상기 셧다운 시퀀스는

상기 컴퓨팅 장치 상에서 임의의 사용자 세션을 해제하는 동작- 상기 해제에 의해 상기 사용자 상태는 더 이상 상기 휘발성 메모리에 유지되지 않음 -과,

상기 해제하는 동작에 후속하여, 상기 휘발성 메모리의 콘텐츠를 비휘발성 메모리에 복사하는 동작- 상기 복사된 콘텐츠는 상기 유지된 시스템 상태는 포함하지만 더 이상 유지되지 않는 상기 사용자 상태는 포함하지 않음 -과,

상기 컴퓨팅 장치의 상기 복수의 하드웨어 구성요소의 적어도 일부분을 파워 다운시키는 동작을 포함하는

컴퓨터 저장 장치.

청구항 9

제 8 항에 있어서,

상기 해제하는 동작은 상기 시스템 상태로부터 사용자 특정 설정을 제거하는 동작을 포함하는

컴퓨터 저장 장치.

청구항 10

제 8 항에 있어서,

상기 셧다운 시퀀스는 상기 파워 다운시키는 동작에 후속하여 또한 시동 명령어를 수신한 것에 응답하여, 상기 복수의 하드웨어 구성요소 중 적어도 상기 일부분을 파워 업시키는 동작을 더 포함하는

컴퓨터 저장 장치.

청구항 11

제 10 항에 있어서,

상기 셧다운 시퀀스는 상기 파워 업시키는 동작에 후속하여, 상기 복사된 콘텐츠를 상기 비휘발성 메모리로부터 상기 휘발성 메모리로 복사하는 동작을 더 포함하는

컴퓨터 저장 장치.

청구항 12

제 8 항에 있어서,

상기 복사되는 콘텐츠를 복사하는 동작은 상기 복사되는 콘텐츠를 포함하는 파일을 검출하는 것에 기초하는

컴퓨터 저장 장치.

청구항 13

제 8 항에 있어서,

상기 콘텐츠를 복사하는 동작은 파일을 생성하는 동작을 포함하는

컴퓨터 저장 장치.

청구항 14

제 8 항에 있어서,

상기 셧다운 시퀀스는 더티 데이터를 상기 비휘발성 메모리에 플러싱하는 동작을 더 포함하는

컴퓨터 저장 장치.

청구항 15

적어도 하나의 프로그램 모듈을 포함하는 컴퓨팅 장치로서,

상기 프로그램 모듈은 셧다운 명령어를 수신한 것에 응답하여 상기 컴퓨팅 장치를 셧다운시키도록 구성된 셧다운 시퀀스를 수행하도록 구성되고, 상기 컴퓨팅 장치는 상기 컴퓨팅 장치의 임의의 사용자 상태와 상기 컴퓨팅 장치의 시스템 상태를 유지하는 휘발성 메모리를 포함하는 복수의 하드웨어 구성요소를 포함하고,

상기 셧다운 시퀀스는

상기 컴퓨팅 장치 상에서 임의의 사용자 세션을 해제하는 동작- 상기 해제에 의해 상기 사용자 상태는 더 이상 상기 휘발성 메모리에 유지되지 않음 -과,

상기 해제하는 동작에 후속하여, 상기 휘발성 메모리의 콘텐츠를 비휘발성 메모리에 복사하는 동작- 상기 복사된 콘텐츠는 상기 유지된 시스템 상태는 포함하지만 더 이상 유지되지 않는 상기 사용자 상태는 포함하지 않음 -과,

상기 컴퓨팅 장치의 상기 복수의 하드웨어 구성요소의 적어도 일부분을 파워 다운시키는 동작

을 수행하도록 더 구성된

컴퓨팅 장치.

청구항 16

제 15 항에 있어서,

상기 해제하는 동작은 상기 시스템 상태로부터 사용자 특정 설정을 제거하는 동작을 포함하는

컴퓨팅 장치.

청구항 17

제 15 항에 있어서,

상기 셧다운 시퀀스는 상기 파워 다운시키는 동작에 후속하여 또한 시동 명령어를 수신한 것에 응답하여, 상기 복수의 하드웨어 구성요소 중 적어도 상기 일부분을 파워 업시키는 동작을 더 포함하는

컴퓨팅 장치.

청구항 18

제 17 항에 있어서,

상기 셧다운 시퀀스는 상기 파워 업시키는 동작에 후속하여, 상기 복사된 콘텐츠를 상기 비휘발성 메모리로부터 상기 휘발성 메모리로 복사하는 동작을 더 포함하는

컴퓨팅 장치.

청구항 19

제 15 항에 있어서,

상기 복사되는 콘텐츠를 복사하는 동작은 상기 복사되는 콘텐츠를 포함하는 파일을 검출하는 것에 기초하는

컴퓨팅 장치.

청구항 20

제 15 항에 있어서,

상기 셧다운 시퀀스는 더티 데이터를 상기 비휘발성 메모리에 플러싱하는 동작을 더 포함하는

컴퓨팅 장치.

발명의 설명

배 경 기 술

[0001] 컴퓨터는 완전 동작(full operation)에서 완전 셧다운(full shutdown)까지 몇 가지 동작 모드를 가진다. 완전 동작에서는, 운영 체제의 실행 부분을 규정하는 소프트웨어가 비휘발성 메모리로부터 휘발성 메모리로 로딩되며, 이로 인해 소프트웨어가 더 빠르게 실행될 수 있다. 컴퓨터는 "시동(startup)" 프로세스를 통해 이러한 완전 동작 모드에 돌입한다. 상기 시동 프로세스는 하드웨어를 구성하고 컴퓨터의 운영 체제를 로딩한다. 시동 프로세스의 일부로서 드라이버가 설치되고 운영 체제 서비스가 시작된다.

[0002] 컴퓨터가 임의의 사용자에 의해 동작될 준비가 되면, 사용자는 컴퓨터에 로그온(log on)할 수 있다. 이 로그온은 로그온하는 사용자 특정 프로파일을 기초로 하는 컴퓨터의 추가 구성을 포함할 수 있다. 그 후 자동으로 또는 사용자 입력에 응답하여, 애플리케이션이 로딩되어, 컴퓨팅 장치의 하드웨어 및 운영 체제 서비스의 능력을 이용하면서, 애플리케이션이 실행될 수 있도록 한다.

[0003] 소프트웨어를 로딩하는 프로세스에서, 운영 체제에 대해서인지 애플리케이션에 대해서인지에 무관하게 메모리가 할당될 수 있고, 소프트웨어의 파라미터에 컴퓨터의 하드웨어 구성 또는 사용자 프로파일을 기초로 하는 값이 할당될 수 있으며, 그 밖의 다른 구성(configuration) 동작이 수행될 수 있다.

[0004] 이들 동작은 컴퓨팅 장치의 "상태(state)"를 확립한다. 사용자가 실행 중인 애플리케이션 또는 운영 체제 서비스와 상호대화하라는 명령어(command)를 제공할 때 동작 상태(operating state)를 규정하는 메모리 및 시스템의 파라미터의 추가적인 변화가 또한 이뤄질 수 있다.

[0005] 완전 셧다운 모드(full shutdown mode)에서는, 컴퓨터의 하드웨어 구성요소로 전력이 공급되지 않는다. 전력이 차단될 때 휘발성 메모리는 정보를 유지하지 않기 때문에 어떠한 소프트웨어 또는 상태 정보도 상기 휘발성 메모리에 저장되지 않는다. 오히려, 완전 동작 모드에 대해 컴퓨터를 재구성하기 위해 추후 사용될 임의의 정보가 비휘발성 메모리에 저장된다.

[0006] 컴퓨터는 셧다운(shutdown)이라고 일컬어지는 프로세스를 통해 셧다운 모드로 돌입한다. 셧다운 동안, 컴퓨터를 재-구성할 필요가 있을 수 있는 임의의 정보가 비휘발성 메모리에 이미 저장되어 있지 않다면, 상기 정보는 비휘발성 메모리에 저장될 수 있다. 비휘발성 메모리로부터 휘발성 메모리로 복사되는 소프트웨어 및 그 밖의 다른 구성 정보는 비휘발성 메모리로 다시 복사되지 않는데, 왜냐하면 뒤 이은 시동 프로세스 후 재생성될 수 있기 때문이다. 그러나 휘발성 메모리 캐시 데이터가 비휘발성 메모리로부터 복사되고 복사된 후 수정된(이따금 "더티" 데이터(dirty data)로 지칭됨) 범위까지, 상기 데이터는 셧다운 동안 비휘발성 메모리로 복사된다.

[0007] 추가 변동이 로그 오프(log off)라고 일컬어진다. 사용자 세션을 지원하는 컴퓨터에서, 사용자는 자신의 기능을

액세스하기 위해 컴퓨터에 로그온할 수 있다. 그러나 셧다운은 사용자를 유효하게 로그 오프하지만, 별도의 로그 오프 프로세스가 수행될 수 있고, 그 이후에 컴퓨터는 파워 다운하지 않는다. 오히려 운영 체제는 로딩된 채 유지되고 또 다른 사용자가 로그온할 준비가 된다. 로그오프 동안 컴퓨터는 사용자 세션을 "해제(break down)" 시킨다. 사용자 세션을 해제하는 것은 사용자에 의해 런칭된 애플리케이션의 종료(closing)와 비휘발성 메모리에 아직 없는 특정 데이터를 저장하는 것을 수반할 수 있다.

[0008] 완전 셧다운 또는 로그 오프에 추가로, 컴퓨터의 하드웨어 구성요소의 일부 또는 전부로의 전력이 차단되는 전력 절약 모드가 있을 수 있다. 때때로 수면 모드(sleep mode)라고 불리는 전력 절약 모드에서, 컴퓨터 프로세서, 네트워크 인터페이스, 및 가능하다면 그 밖의 다른 구성요소로의 전력이 차단된다. 그러나 휘발성 메모리에 대한 전력은 유지된다. 이러한 방식으로 컴퓨터의 부트(boot) 또는 뒤 이은 동작 중에 형성된 임의의 상태 정보가 휘발성 메모리에 유지된다. 프로세서로 다시 전력이 공급될 때, 수면 모드로 돌입할 때 차단 상태로 유지되는 상태에서 동작이 재개될 수 있다.

[0009] 때때로 추가 모드는 휴면 모드(hibernate mode)라고 불린다. 컴퓨터는 휴면이라고 불리는 프로세스를 통해 이 모드에 돌입한다. 휴면 동안, 컴퓨터의 동작 상태를 캡처하는 파일이 생성되고 비휘발성 메모리, 일반적으로 하드 디스크에 저장된다. 휴면으로부터 재개하는 프로세스 동안, 이 파일이 디스크로부터 읽히고 컴퓨터의 상태를 휴면 시점에서 존재했던대로 재-확립하기 위해 사용될 수 있다. 휴면으로부터의 재개는 휴면 시점에서 존재했던 동작 동안 설정된 소프트웨어 또는 파라미터의 메모리 복사본을 휘발성 메모리에 복원하여, 어떠한 사용자 상태라도 복원될 수 있도록 한다.

[0010] 몇 가지 이유에서 휴면으로부터 재개하는 것이 완전 시동을 수행하는 것보다 빠를 수 있다. 한 가지 이유는 휴면 파일 내 상태 정보를 휘발성 메모리로 복사하는 것이 시동 프로세스의 단계, 가령, CPU 소비, 장치 초기화 및 부트 동안 수행되어야 할 그 밖의 다른 많은 유형의 일을 실행하는 데 소요되는 시간을 피하면서 완전 시동 프로세스의 결과를 재-생성하기 때문이다. 추가로, 시동 동안 액세스되는 정보가, 아마도 운영 체제 내 수 만개에 달하는 구성요소일 수 있는 것을 로딩 및 구성하도록 액세스되는 여러 다른 구성요소를 나타내는 여러 다른 파일에 저장된다. 이들 구성요소 및 이들을 구성하기 위해 액세스되는 정보는 하드 디스크에 걸쳐 랜덤하게 분포될 수 있다. 하드 디스크 드라이브, 및 그 밖의 다른 임의의 형태의 고용량 저장장치는 순차 데이터 액세스에서 가장 높은 효율을 보이기 때문에, 랜덤 분포된 데이터를 액세스하는 것은 상당한 디스크 액세스 시간을 포함할 수 있으며, 이는 긴 시동 프로세스를 초래한다. 이와 달리, 휴면 파일을 읽을 때는 액세스 시간이 짧은데, 왜냐하면 상기 파일 내 정보가 디스크에 순차적으로 저장될 수 있기 때문이다.

[0011] 휴면으로부터 재개하는 것과 시동 간의 추가 차이점은 휴면 및 그 후 재개가, 컴퓨터가 휴면됐을 때의 컴퓨터의 사용자에 대한 임의의 사용자 상태를 포함해 컴퓨터의 완전 상태를 복원한다는 것이다. 이와 달리, 사용자가 로그온할 때까지, 일반적으로 시동은 임의의 사용자에 대해 컴퓨터를 구성할 것이다. 그 후 특정 사용자는 로그온하거나 그 밖의 다른 방식으로 그들 스스로 컴퓨터를 구성하기 위한 동작을 취할 수 있다. 이러한 이유로, 일반적으로 휴면은 컴퓨터로부터 잠시 동안 떨어져 있지만 컴퓨터로 복귀하려는 사용자에 의해 선택된다. 일반적으로 셧다운은 더 긴 시간 동안 컴퓨터로부터 떨어져 있고, 아마도 컴퓨터로 복귀하지 않을 사용자 또는 자신이 복귀하기 전에 타 사용자가 컴퓨터를 이용할 수 있음을 예상하는 사용자에 의해 사용된다.

발명의 내용

과제의 해결 수단

[0012] 사용자 경험을 개선하기 위해, 컴퓨터는 휴면 모드에 진입함으로써 셧다운(shutdown)하기 위한 사용자 명령어(command)에 응답하도록 구성될 수 있다. 이러한 컴퓨터는 사용자가 컴퓨터를 시동(startup)하기 위한 명령을 제공한 후 더 빠르게 사용자에 의해 동작될 준비가 될 수 있다. 더 빠르게 컴퓨터가 사용자의 기대(user's expectation)와 일치하는 상태로 동작할 준비가 되게 하기 위해, 휴면 파일(hibernation file)이 사용자 기대를 구현하는 목표 상태(target state)를 포착한다. 셧다운 명령어에 응답하여, 컴퓨터는 셧다운 프로세스의 단계들 중 일부만 수행함으로써 휴면 전에 이 목표 상태를 생성한다. 수행되는 단계들은 컴퓨터가 목표 상태가 되게 할 수 있으며, 상기 목표 상태는 운영 체제가 로딩된 채 유지되지만 사용자 세션은 해제된 상태에 대응한다.

[0013] 시동 명령의 수신이 있으면, 컴퓨터 시스템은, 소프트웨어를 로딩하고 구성함으로써 동작 상태를 생성하기 보다는, 상기 휴면 파일을 휘발성 메모리로 복사함으로써 상기 목표 상태를 재-생성한다. 그 후, 컴퓨터는 시동 시

퀀스의 일부분만 수행할 수 있다. 이러한 일부분은 운영 체제가 로딩된 후 통상적으로 시동 시퀀스 동안 발생할 동작을 포함할 수 있다. 이들 단계는, 예를 들어, 사용자 로그온(log on)을 수행하고 사용자 상태(user state)를 규정하는 애플리케이션을 로딩하기 위해 사용자와 상호대화(interact)하는 단계를 포함할 수 있다.

[0014] 일부 실시예에서, 셧다운을 가리키는 사용자 명령어에 응답하여 조건부 프로세싱이 수행될 수 있다. 상기 컴퓨팅 장치는, 예를 들어, 컴퓨팅 장치가 완전 셧다운이 요구되는 동작 상태에 있는지 여부, 또는 다음 번 시동 명령어에 응답하여 사용될 휴면 파일을 생성하는 것이 적절한지 여부를 결정할 수 있다.

[0015] 이러한 상태는 많은 방식들 중 임의의 방식으로 식별될 수 있으며, 가령, 일부 설치된 구성요소의 구성 설정이 변경되었고, 완전 시동 시퀀스의 일부로 구성요소가 다시 로딩될 때까지 적용되지 않을 것이라고 결정함으로써, 식별될 수 있다. 또는, 완전 셧다운을 필요로 하는 것으로 애플리케이션 구성요소가 등록되게 하는 프로그램 인터페이스가 제공될 수 있다.

[0016] 이러한 조건이 검출되는 경우, 컴퓨팅 장치가 완전히 파워 다운(power down)될 때까지 종래의 셧다운 프로세싱이 수행될 수 있다. 이러한 조건이 검출되지 않은 경우, 셧다운 시퀀스는 컴퓨팅 장치가 휴면 파일이 만들어질 수 있는 목표 상태로 될 때까지 수행될 수 있다.

[0017] 일부 실시예에서, 사용자 시동 명령어에 응답하여 조건부 프로세싱이 수행될 수 있다. 상기 조건부 프로세싱은 휴면 파일이 존재하는지 여부를 결정하는 것을 포함할 수 있다. 휴면 파일이 존재하는 경우, 휴면 파일이 생성된 시점과 시동 명령어가 수신된 시점 사이에 컴퓨팅 장치의 목표 상태가 변경됐을 가능성이 있는지 여부에 대해 추가 체크가 이뤄질 수 있다. 상태 변경을 야기했을 수 있는 이벤트가 검출된 경우, 컴퓨팅 장치는 완전 시동 시퀀스를 수행할 수 있다.

[0018] 상기의 내용은 본 발명의 비-제한적 개요이며, 본 발명은 첨부된 청구항에 의해 규정된다.

도면의 간단한 설명

[0019] 첨부된 도면은 실측 비율로 그려지지 않았다. 도면에서, 다양한 도면들 간에 도시된 동일하거나 거의 동일한 구성요소 각각은 유사한 도면 부호로 나타내어 진다. 간결성을 위해, 매 도면에서 모든 구성요소가 라벨링된 것은 아니다. 도면의 설명은 다음과 같다.

도 1은 컴퓨팅 장치에서의 시동 시퀀스를 도시하는 개념 블록도이다.

도 2는컴퓨팅 장치에서의 휴면 시퀀스로부터의 재개를 도시하는 기능 블록도이다.

도 3은 본 발명의 일부 실시예에 따르는 고속 시동 시퀀스를 도시하는 기능 블록도이다.

도 4는 본 발명의 일부 실시예에 따르는 시동 명령어에 응답하기 위한 컴퓨터의 동박 방법을 도시하는 흐름도이다.

도 5는 본 발명의 일부 실시예에 따르는 셧다운 명령어에 응답하도록 컴퓨팅 장치를 동작시키기 위한 방법의 흐름도이다.

도 6은 본 발명의 일부 실시예에 따라 조건부로 실행될 수 있는 시동 시퀀스의 일부분의 흐름도이다.

도 7은 셧다운이 있을 시 컴퓨팅 장치의 서로 다른 거동을 야기하는 명령어들을 사용자가 선택할 수 있게 해주는 그래픽 사용자 인터페이스의 일부분의 스케치이다.

도 8은 본 발명의 실시예가 동작될 수 있는 하나의 환경을 도시하는 예시적 컴퓨팅 장치의 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0020] 본 발명자는 컴퓨팅 장치의 사용자의 경험이, 컴퓨팅 장치의 셧다운 및/또는 시동 시퀀스의 일부분과 조합되는 휴면 파일(hibernation file)의 사용을 통해 개선될 수 있음을 알았다. 이러한 파일은 셧다운 후 선택적으로 선택되고 시동 후 선택적으로 사용되어, 컴퓨팅 장치의 성능이 사용자 기대(user expectation)에 부합하도록 할 수 있다. 휴면 파일이 생성되거나 사용될 때라도, 종래의 셧다운 또는 시동 시퀀스의 일부분이 수행될 수 있다.

[0021] 사용자 기대와 일치되는 컴퓨팅 장치의 동작을 제공하기 위해, 컴퓨팅 장치를 목표 상태(target state)로 두는 컴퓨팅 장치의 종래의 셧다운 시퀀스의 일부분과 조합하여 휴면이 사용될 수 있다. 이들 부분은, 셧다운 명령어

의 수신 후, 사용자 세션을 해제(break down)하는 동작을 포함할 수 있다. 이에 추가로, 셧다운 명령어에 응답하는 일부로서, 사용자 세션이 해제된 후 휘발성 메모리에 유지되지만 비휘발성 메모리에 유지되지 않도록 의도된 정보가 비휘발성 메모리로 이동된다. 예를 들어, 종래의 셧다운 동안 수행되는 동작을 모방하는 종래의 캐시 플러쉬(cache flush) 동작이 수행될 수 있다.

[0022] 이와 달리, 시동(startup) 명령어의 프로세싱 후, 휴면으로부터의 재개가 시동 시퀀스의 일부분과 함께 수행될 수 있다. 상기 시퀀스는 운영 체제가 로딩되고 동작될 준비가 된 후 발생하는 시동 시퀀스의 임의의 부분을 포함할 수 있다. 예를 들어, 상기 시동 시퀀스의 일부분은 사용자 로그온 및 애플리케이션의 로딩을 포함할 수 있다.

[0023] 추가로, 사용자 기대와 일치하는 동작을 제공하기 위해, 셧다운 또는 시동의 일부로서의 휴면 파일의 생성 또는 사용이 동적으로 결정된 이벤트에 따라 조절(condition)될 수 있다. 동작 세션(operating session) 중에 구성요소가 재구성되어, 구성요소가 다음 번 로딩될 때까지 구성(configuration) 변화가 적용되지 않는 시나리오에서, 어떠한 휴면 파일도 생성되지 않을 수 있다. 사용자로부터의 다음 번 시동 명령어에 응답하여, 컴퓨터는 어떠한 휴면 파일도 이용 가능하지 않다고 검철하고 운영 체제를 재로딩(reloading)함으로써 목표 상태를 생성할 것이다. 대안적으로 또는 추가적으로, 운영 체제는 한 인터페이스를 제공할 수 있는데, 상기 인터페이스를 통해 그 밖의 다른 구성요소가 효과적으로 기능하기 위해 완전 셧다운 또는 시동을 필요로 한다고 알리기 위해 등록할 수 있다. 실행 구성요소가 등록될 때, 완전 셧다운 시퀀스가 셧다운 명령어에 응답하여 수행될 수 있다.

[0024] 추가로, 사용자 기대에 일치하여 동작하기 위해, 일부 실시예에서, 사용자 인터페이스가 제공될 수 있고, 상기 사용자 인터페이스를 통해 사용자는 종래의 셧다운을 수행할지, 또는 목표 상태가 생성되고 그 후 휴면 프로세스가 수행되는 수정 셧다운을 수행할지를 특정할 수 있다. 이러한 사용자 인터페이스는 종래의 셧다운과 휴면을 포함하는 수정 셧다운 시퀀스에 대한 개별 옵션을 제공할 수 있다. 컴퓨팅 장치는 종래의 셧다운 명령어로서 라벨링된 입력에 응답하여 수정 셧다운 시퀀스를 조건부로 불러올 수 있다(invoke). 사용자가 종래의 셧다운을 특정할 수 있는 상기 인터페이스를 통해 개별 명령어 옵션이 제공될 수 있다.

[0025] 다시 도 1로 돌아와서, 완전 시동 시퀀스의 기능 블록도가 도시된다. 도 1은 본 발명의 실시예에 따라 동작하도록 적응될 수 있는 컴퓨팅 장치(100)의 기능 블록도이다.

[0026] 이 예시에서, 컴퓨팅 장치(100)는 휘발성 메모리(120)를 포함한다. 휘발성 메모리(120)는 DRAM 또는 그 밖의 다른 임의의 적합한 메모리 구성요소를 이용해 구현될 수 있다. 컴퓨팅 장치(100)에 의해 수행되는 시동 시퀀스는 컴퓨팅 장치(100)로 하여금 해당 분야에 잘 알려진 바와 같은 컴퓨팅 동작을 수행하게 하는 상태 정보를 휘발성 메모리(120) 내에 생성하는 것을 포함한다.

[0027] 이 예에서, 상기 상태 정보는 2개의 부분, 즉, 사용자 상태 정보(130)와 시스템 상태 정보(140)을 갖는 것으로 설명된다. 일반적으로 시스템 상태 정보(140)는 임의의 사용자에 의해 동작되기 위한 컴퓨팅 장치(100)를 구성하는 상태 정보를 나타낸다. 이와 달리, 사용자 상태 정보(130)는 컴퓨팅 장치(100)가 동작되거나 특정 사용자에 의해 동작되도록 구성될 때 생성될 수 있는 상태 정보를 나타낸다.

[0028] 해당 분야에 공지된 시동 프로세스에 따라 시스템 상태 정보(140) 및 사용자 상태 정보(130)는 휘발성 메모리(120)에 생성될 수 있다. 도 1은 단순화된 개념 방식으로, 종래의 시동 시퀀스의 단계들을 도시한다. 예를 들어, 컴퓨팅 장치(100)가 켜질 때(power on) 또는 시동을 알리는 그 밖의 다른 명령어가 제공될 때, 이러한 시퀀스가 개시될 수 있다.

[0029] 컴퓨팅 장치(100)는 해당 분야에 공지된 것과 같은 구성요소들을 포함할 수 있다. 이들 구성요소는 프로세서(110)를 포함할 수 있다. 해당 분야에 공지된 것처럼, 프로세서(110)는 마이크로프로세서 또는 마이크로프로세서 또는 프로세서 코어의 집합으로서 구현될 수 있다. 본원에 기재된 동작은 소프트웨어 명령(software instruction)을 실행한 프로세서(110)의 결과일 수 있다.

[0030] 추가로, 컴퓨팅 장치(100)는 복수의 유형의 컴퓨터 저장 매체를 포함할 수 있다. 이 경우, 이들 유형으로는 휘발성 메모리 및 비휘발성 메모리가 있다. 이 예에서, 휘발성 메모리(120)가 도시된다. 다양한 유형의 정보가 비휘발성 메모리(150 및 152)에 저장된다. 부트 메모리(Boot memory)(154)가 또한 비휘발성 메모리이다. 상이한 물리적 장치가 비휘발성 메모리(150 및 152) 및 부트 메모리(154)를 구현하도록 사용될 수 있다. 예를 들어, 비휘발성 메모리(150)는 디스크, 가령, 스피닝 하드 디스크(spinning hard disk), 또는 솔리드 스테이트 드라이브(solid state drive)일 수 있다. 비휘발성 메모리(152)는 디스크와 유사할 수 있고, 비휘발성 메모리(150)를 구현하기 위해 사용되는 것과 동일한 디스크이거나, 동일한 디스크 상의 서로 다른 파티션(partition) 또는 전체

적으로 서로 다른 디스크일 수 있다.

[0031] 마찬가지로 비휘발성 메모리(154)는 비휘발성 메모리(150 및 152)를 구현하기 위해 사용되는 것과 동일한 장치의 일부분일 수 있다. 그러나 도시된 실시예에서, 비휘발성 메모리(154)는 프로세서(110)로 연결된 비휘발성 메모리 칩일 수 있다. 따라서 도 1은 단지 메모리 아키텍처의 하나의 예를 나타내고, 임의의 적합한 메모리 아키텍처가 사용될 수 있음을 알아야 한다.

[0032] 이 예에서, 비휘발성 및 휘발성 메모리가 도시된다. 이러한 구성은 전통적인 컴퓨터 아키텍처를 나타낸다. 그러나 반드시 특정 아키텍처가 사용되어야 하는 것은 아니다. 오히려, 휘발성 메모리(120)는 동작 메모리의 일례이다. 컴퓨팅 장치(100)의 동작 중에, 프로세서(110)는 휘발성 메모리(120)로부터의 동작을 수행하기 위한 소프트웨어 및 데이터를 주로 액세스할 수 있다. 이 메모리는 비교적 고속일 수 있어서 상기 동작이 빠르게 프로세서(110)에 의해 수행될 수 있도록 한다.

[0033] 이와 달리, 비휘발성 메모리, 가령, 비휘발성 메모리(150 및 152)는 많은 양의 데이터를 저장할 수 있지만 휘발성 메모리(120)보다 느리게 동작할 수 있다. 일반적으로 정보를 이러한 비휘발성 메모리에 저장하는 비용은 정보를 휘발성 메모리(120)에 저장하는 비용에 비해 비교적 작다. 고속의 동작과 함께 비용 효율성을 이루기 위해, 정보가 비휘발성 메모리와 휘발성 메모리 사이에서 이동될 수 있다. 이러한 이동은 컴퓨팅 장치(100)의 원하는 동작을 지원하는 상태를 휘발성 메모리(120) 내에 생성하기 위해 수행된다.

[0034] 컴퓨터 시스템의 그 밖의 다른 구성요소가 존재할 수 있지만, 간결성을 위해 생략된다. 그 밖의 다른 실시예에서 존재할 수 있는 구성요소의 더 많은 상세사항이 도 8과 관련하여 이하에서 제공된다. 그러나 시동 프로세스를 설명하기 위해 도 1의 단순화된 도시가 적절하다.

[0035] 시동 명령어에 응답하여, 프로세서(110)는 부트 메모리(154) 내 명령을 액세스하고 실행시킬 수 있다. 부트 메모리(154)는 프로세서(110)가 비휘발성 메모리(150 및 152)를 액세스하고, 이들 메모리에 저장된 소프트웨어 및 데이터를 기초로 하여 휘발성 메모리(120)에 적절한 상태를 생성할 수 있도록 하는 명령을 포함할 수 있다.

[0036] 부트 메모리(154) 내 명령은 프로세서(110)로 하여금 비휘발성 메모리(150)로부터 소프트웨어를 로딩하게 할 수 있다. 소프트웨어 구성요소를 로딩하는 것의 일부로서, 프로세서(110)는 소프트웨어 명령을 휘발성 메모리(120)로 전달하고, 상기 휘발성 메모리에서 소프트웨어가 실행될 수 있다. 그러나 소프트웨어를 로딩하는 것은 그 밖의 다른 동작, 가령, 일부 구성요소의 실행을 포함할 수 있다.

[0037] 휘발성 메모리(120)에서의 일부 구성요소의 실행은 소프트웨어를, 소프트웨어가 저장된 상태에서, 사용되거나 그 밖의 다른 구성요소가 비휘발성 메모리로부터 휘발성 메모리(120)로 전달되게 하는 상태이도록 변환할 수 있다. 소프트웨어의 로딩 프로세스에서, 프로세서(110)는 비휘발성 메모리(152)에 저장된 데이터 또는 그 밖의 다른 정보를 기초로 소프트웨어를 구성할 수 있다. 상기 정보는, 예를 들어, 컴퓨팅 장치(100)에 설치되는 하드웨어 구성요소에 대한 정보를 포함할 수 있다. 따라서 도 1에서, 시동 프로세스의 제 2 및 제 3 단계는 비휘발성 메모리(150)로부터 소프트웨어, 그리고 비휘발성 메모리(152)로부터 데이터를 획득하는 것일 수 있다.

[0038] 이 프로세스에서 로딩되는 제 1 소프트웨어는 시스템 상태(140)를 확립할 수 있다. 초기에 로딩된 소프트웨어는 시스템 상태(140)에 하드웨어 구성요소를 제어하는 드라이버(146)를 추가할 수 있다. 드라이버를 로딩하기 전에, 컴퓨팅 장치(100)와 연관된 하드웨어 구성요소가 식별되고 적절한 드라이버가 선택될 수 있다. 드라이버가 설치되면 운영 체제 서비스, 및 그 밖의 다른 구성요소는 상기 드라이버를 통해 제어되는 장치와 상호대화할 수 있다.

[0039] 그 후 운영 체제 서비스(142)가 로딩될 수 있다. 이러한 서비스의 한 가지 예는 파일 관리자(file manager)(144)이다. 운영 체제 서비스 및 애플리케이션을 실행하는 것이 파일에 따라 조직된 비휘발성 메모리 내 데이터를 액세스할 수 있도록 파일 관리자(144)는 휘발성 메모리에서 데이터를 조직할 수 있다. 운영 체제에 의해 제공되는 그 밖의 다른 서비스는 사용자 인터페이스와 상호대화하는 것, 네트워크 연결을 확립하는 것, 또는 정보를 프린터로 전송하는 것을 포함할 수 있다. 그러나 특정 운영 체제 서비스(142)로 본 발명이 제한되는 것은 아니다.

[0040] 덧붙이자면, 시스템 상태(140)를 확립하는 프로세스 동안, 프로세서(110)는 시스템 상태 데이터(148)를 저장할 수 있다. 이러한 데이터는 비휘발성 메모리, 가령, 비휘발성 메모리(152)로부터 복사되거나, 소프트웨어 구성요소의 실행에 의해 생성될 수 있다. 예를 들어, 프로세서(110)가 컴퓨팅 장치(100) 내에 설치된 장치를 발견(discover)하는 명령을 실행할 때 데이터가 생성될 수 있다. 구체적 예를 들면, 특정 네트워크 인터페이스 카드를 발견하면, 프로세서(110)는 시스템 상태 데이터(148)의 일부로서 네트워크 인터페이스 카드의 유형 또는 능

력을 기록할 수 있다. 그 후 이 데이터는 컴퓨팅 장치의 동작 중에 네트워크 인터페이스 카드와의 상호대화를 제어하기 위해 사용될 수 있다. 그러나 시스템 상태 데이터(148)로서 저장된 특정 데이터가 본 발명의 필수 핵심은 아니다.

[0041] 특정 운영 체제 서비스(142) 및 시스템 상태 정보(140)에서 생성되는 시스템 상태 데이터(148)에 무관하게, 상기 시스템 상태 정보(140)가 생성될 때, 컴퓨팅 장치(100)는 사용자에 의해 동작될 준비가 될 수 있다. 따라서 시동 시퀀스가, 때때로 사용자 로그온(user log on)이라고 지칭되는 프로세스를 지속시킬 수 있다. 사용자 로그온의 일부로서, 특정 사용자가 식별되고 추가 상태 정보가 비휘발성 메모리(120)에 생성되어, 컴퓨팅 장치(100)로 하여금 상기 사용자에 대한 동작을 수행하게 할 수 있다. 이 예에서, 사용자 상태 정보(130)는 애플리케이션 명령(132)과 사용자 상태 데이터(134)를 포함하는 것으로 나타난다.

[0042] 운영 체제 구성요소를 나타내는 명령 및 시스템 상태를 나타내는 데이터를 이용하는 경우처럼, 애플리케이션 명령(132)은 휘발성 메모리(150)에 저장되는 소프트웨어 기반 메모리로 로딩될 수 있다. 그러나 소프트웨어를 로딩하는 프로세스는 동작되도록 소프트웨어 또는 컴퓨팅 장치를 적절하게 구성하는 기능을 실행시키는 것을 수반할 수 있다. 상기 구성은 시스템 상태 데이터(148) 또는 사용자 상태 데이터(134)에 따라 달라질 수 있다.

[0043] 하나의 예를 들면, 웹 브라우저를 구현하는 애플리케이션 명령이 로딩되면, 프로세서(110)가 비휘발성 메모리(152) 또는 사용자 상태 데이터(134)로부터, 사용자가 "즐겨찾기(favorites)"로 식별한 특정 웹 사이트를 식별하는 사용자 데이터를 나타내는 정보를 액세스할 수 있다. 이 예에서, 사용자 상태 데이터(130)를 확립하는 것은 사용자 선호에 따라 실행되도록 웹 브라우저를 구성하며, 이는 컴퓨팅 장치(100)에 로그온한 특정 사용자에 대해 커스텀화된 즐겨찾기의 리스트를 제공하는 것을 포함할 수 있다.

[0044] 사용자 로그온이 완료되면, 사용자는 컴퓨팅 장치(100)와 상호대화할 수 있다. 이들 상호대화는 추가 소프트웨어가 로딩되거나 일부 로딩된 애플리케이션이 종료되는 것을 야기할 수 있다. 추가로, 사용자 상호대화는 파라미터를 설정하거나 사용자 상태(130) 또는 시스템 상태(140)를 변경할 수 있는 또 다른 동작을 취할 수 있다. 이들 상호대화는 사용자가 세션을 종료하라는 의도를 나타내는 명령어를 입력할 때까지 계속될 수 있다.

[0045] 세션은 여러 방식들 중 하나로 종료될 수 있다. 예를 들어, 사용자가 컴퓨팅 장치(100)와의 상호대화 세션을 완료할 때, 사용자는 컴퓨팅 장치(100)를 로그오프(log off) 및/또는 셧다운할 수 있다. 로그오프에 의해 사용자 상태 정보(130)가 메모리(120)에서 더 이상 이용 가능하지 않도록 사용자 세션이 해제될 수 있다. 로그오프 시퀀스의 일부분은 시스템 상태(140)로부터 사용자 특정 설정을 제거하는 것을 수반할 수 있다. 이러한 방식으로, 이전 사용자에 의해 생성된 상태 정보에 의해 영향받거나 상기 상태 정보를 액세스할 수 있지 않으면서, 두 번째 사용자가 컴퓨팅 장치(100)에 로그온할 수 있다. 때때로 이 결과를 달성하기 위한 동작은 사용자 세션의 해제로서 기재될 수 있다.

[0046] 메모리(120)로의 전력이 유지될 수 있기 때문에, 로그오프 후에 시스템 상태(140)가 유지될 수 있다. 이와 달리, 셧다운은 사용자 상태(130)와 시스템 상태(140) 모두 휘발성 메모리(120)로부터 제거되게 할 수 있다. 휘발성 메모리(120)로의 전력이 꺼지기 때문에, 셧다운 시퀀스의 종료 부분에서의 휘발성 메모리(120) 내 임의의 정보가 소실될 것이다. 따라서 상기 상태를 재-생성하기 위해 필요한 임의의 정보는, 비휘발성 메모리에 이미 저장된 것이 아니라면, 상기 비휘발성 메모리로부터 제거될 수 있다.

[0047] 비휘발성 메모리로부터 생성된 임의의 정보를 반환할 어떠한 필요도 없기 때문에, 로그오프 및/또는 셧다운 시퀀스가 반드시 시동 시퀀스의 반전(reverse)인 것은 아니다. 상기 동일한 정보가 뒤 이은 시동 후에 다시 생성될 수 있다. 그러나 세션 중에 동적으로 생성되었고 비휘발성 메모리 내 정보로부터 재-생성될 수 없는 사용자 상태(130)의 일부분은, 로그오프 또는 셧다운 동작의 일부로서, 비휘발성 메모리에 기록될 수 있다. 마찬가지로, 셧다운되면, 시동 시퀀스의 재-실행 시 재-생성될 수 없는 시스템 상태 데이터(148)의 일부분이 셧다운 시퀀스의 일부로서 비휘발성 메모리로 전달될 수 있다.

[0048] 예를 들어, 시스템 상태 데이터(148)는 비휘발성 메모리(152)에 저장된 데이터 아이템의 작업 복사본(working copy)로서 기능하도록 의도되어 캐시에 포함될 수 있다. 캐시는 비휘발성 메모리에 유지되어야 할 정보의 복사본을 휘발성 메모리에 확립함으로써 컴퓨팅 장치(100)의 동작을 빠르게 만든다. 더 고속의 휘발성 메모리 위치 내 정보를 읽거나 쓰는 것이, 비휘발성 메모리 내 동일한 데이터를 액세스하는 것에 비교하여, 컴퓨팅 장치의 동작의 속도를 빠르게 만든다.

[0049] 비휘발성 메모리 내 데이터의 복사본이 변경될 때, 상기 복사본은 비휘발성 메모리 내 대응하는 데이터와 더 이상 일치되지 않는다. 캐시 내 데이터는 "더티(dirty)" 데이터라고 불리운다. 비휘발성 메모리를 캐시 내 복사본

과 동기하도록 유지하기 위해, 이따금 더티 데이터가 비휘발성 메모리로 복사된다. 보통, 컴퓨터가 달리 바쁘지 않을 때 더티 데이터는 다시 복사된다.

[0050] 그러나 더티 데이터의 복사를 지연시키는 것이 셧다운 시 캐시 내 데이터가 비휘발성 메모리 내에 있는 것과 일치하지 않을 가능성을 만든다. 불일치를 피하기 위해, 컴퓨팅 장치(100)를 셧다운하기 전에, 때때로 더티 데이터 플러쉬(flushing)라고 일컬어지는 동작이 수행될 수 있다. 이 동작 동안, 더티 데이터는 비휘발성 저장장치로 복사된다.

[0051] 도 1에 도시된 시동 시퀀스는 사용자에 의한 동작을 위해 컴퓨팅 장치(100)를 구성하기 때문에 바람직하더라도, 일부 경우, 시동 시퀀스가 좌절의 원인일 수 있다. 운영 체제 및 사용자가 원하는 애플리케이션은 수천 또는 수만에 달하는 구성요소를 집합적으로 포함할 수 있다. 따라서 시동 시퀀스는 비휘발성 메모리(150 및 152)로부터의 복수의 읽기 동작을 수반할 수 있다. 일반적으로 이들 메모리가 느리게 동작하기 때문에, 전체 프로세스는 비교적 느릴 수 있다. 덧붙이자면 시동 시퀀스는 저장-관련 동작외의 시간 소모 동작을 수반할 수 있다. 추가로 예를 들어, CPU 또는 장치 초기화에 의한 계산에 시간이 소요될 수 있다.

[0052] 도 2는 휘발성 메모리에 상태 정보를 생성하기 위한 대안적 접근법을 도시한다. 도 2는 휴면 시퀀스로부터의 재개의 일부로서, 상태 정보가 휘발성 메모리(120)에 생성되는 동작의 시퀀스 동안 컴퓨팅 장치(100)를 도시한다.

[0053] 휴면은 상태 정보를 휘발성 메모리로부터 비휘발성 메모리로 복사함으로써 생성될 수 있는 동작 모드이다. 이러한 상태 정보는 임의의 적합한 방식으로 구성될 수 있다. 도 2에 도시된 실시예에서, 상기 상태 정보는 비휘발성 메모리(152) 내 휴면 파일(210)에 저장되는 것으로 도시된다. 휴면 동안 프로세서(110)는 상태 정보, 가령, 사용자 상태 정보(130)와 시스템 상태 정보(140)를 휴면 파일(210)로 복사할 수 있다. 그 후 컴퓨터 시스템(100)의 구성요소들 모두 또는 일부로의 전력을 차단함으로써, 휴면 모드에 돌입된다. 전력이 차단될 때, 휘발성 메모리(120) 내 상태 정보가 소실된다. 그러나 상기 상태 정보는, 휴면 파일을 휘발성 메모리로 복사함으로써 휴면으로부터의 재개로서 재-생성될 수 있다.

[0054] 따라서 도 2는 휴면 시퀀스로부터의 재개가 부트 메모리(154)에 저장된 명령을 액세스하는 프로세서(110)에 의한 도 1에 도시된 시동 시퀀스와 유사하게 시작될 수 있다. 이들 명령에 의해 프로세서(110)는 휴면 파일(210)의 존재 여부를 체크할 수 있다. 이 예에서, 휴면 파일(210)을 검출하면, 프로세서(110)는 휴면 파일(210)의 내용을 휘발성 메모리(120)로 복사한다. 이러한 복사는 직접 복사를 수반하거나 정보를 복사되는 것과 같은 임의의 방식으로 변환하기 위해 프로세싱하는 것, 가령, 파일을 압축해제(decompressing)하는 것을 수반할 수 있다. 프로세싱이 프로세싱의 일부로서 수행되는지 여부에 무관하게, 최종 결과는 상태 정보 복원일 것이다. 상태 정보가 복원되면, 사용자는 휴면 시점에서 중단(interrupt)되었던 컴퓨팅 세션을 재개할 수 있다. 시스템 상태 데이터(148)와 사용자 상태 데이터(134) 모두 휘발성 메모리(120)로 반환될 것이다. 덧붙이자면, 마찬가지로, 애플리케이션(132), 운영 체제 서비스(142), 및 드라이버(146)가 휘발성 메모리(120)로 반환되고, 실행될 준비가 될 것이다.

[0055] 종종, 휴면으로부터의 재개가 도 1과 관련하여 설명된 완전 시동 시퀀스(full startup sequence)를 수행하는 것보다 빠를 것이다. 휴면으로부터의 재개 동안 결국 동일한 양의 정보가 휘발성 메모리(120)로 배치될 수 있지만, 상기 정보를 파일로부터 복사하는 것이 소프트웨어 및 구성 데이터를 로딩함으로써 상기 정보를 생성하는 것보다 빠를 수 있다.

[0056] 그러나 휴면 모드에 돌입하고, 그 후 휴면으로부터 재개하는 것이 항상 셧다운을 수행하고 그 후 시동 시퀀스를 수행하는 것을 적절하게 대체하는 것은 아니다. 출원인은 사용자 명령어에 응답하여 휴면을 수행하고, 그 후 컴퓨팅 장치를 시동하기 위한 사용자로부터의 명령어에 응답하여 휴면으로부터 재개하는 것이 사용자의 기대를 충족시키는 컴퓨팅 장치의 동작을 야기하지 않을 수 있음을 알았다.

[0057] 본 발명자는 기존 사용자 기대를 낮추지 않으면서 더 빠른 동작 경험을 제공하기 위한 방법을 식별했다. 도 3은 컴퓨팅 장치(100)가 휴면을 셧다운 시퀀스에 조건부로 포함시킬 수 있는 블록도를 도시한다. 덧붙여, 컴퓨팅 장치는 시동 시퀀스에 휴면으로부터의 재개를 조건부로 포함시킬 수 있다.

[0058] 도 3에 도시된 실시예에서, 컴퓨팅 장치(100)는 비휘발성 메모리(152)로 복사되는 상태 정보를 포함하도록 나타난다. 이 실시예에서, 상태 정보는 휴면 파일(310)로서 포맷팅된다. 휴면 파일(310)은 해당 분야에 공지된 바와 같은 휴면 파일의 형태를 가질 수 있다. 비휘발성 메모리에 상태 정보를 저장하기 위해 임의의 적합한 포맷이 사용될 수 있음을 알아야 한다.

[0059] 휴면 파일(210)에 저장된 정보와 달리, 휴면 파일(310)은 시스템 상태(140)를 포함한다. 사용자 상태(130)는 휴

면 파일(310)의 일부로서 저장될 필요가 없지만, 일부 실시예에서, 사용자 상태의 일부분이 저장될 수 있다. 따라서 사용자가 시동 명령어를 컴퓨팅 장치(100)로 제공할 때, 프로세서(110)는, 도 2에 도시된 동작 모드에서 발생하는 것과 유사하게, 부트 메모리(154)로부터의 명령을 실행하기 시작할 수 있다. 휴면 파일(310)의 존재를 검출하면, 프로세서(110)는 휴면 파일(310)의 내용을 휘발성 메모리(120)로 복사할 수 있다. 이 복사는 휘발성 메모리(120)에 시스템 상태(140)를 재-생성한다.

[0060] 이 상태는, 운영 체제 소프트웨어가 로딩된 후, 그러나 사용자 로그온이 발생하기 전인 도 1에 도시된 시동 시퀀스 동안의 컴퓨팅 장치(100)의 상태를 모방할 수 있다. 따라서 휘발성 메모리(120) 내 상태 정보의 생성을 완료하기 위해, 프로세서(110)는 시스템 상태가 생성된 후 발생하는 도 1과 관련하여 앞서 기재된 시동 시퀀스의 단계들을 수행할 수 있다. 이 경우, 상기 동작들은 애플리케이션 명령(132)을 로딩하고, 비휘발성 메모리(150)로부터의 소프트웨어 명령을 읽고, 비휘발성 메모리(152) 내 데이터를 기초로 하여 이를 구성함으로써, 사용자 상태 데이터(134)를 생성하는 것을 포함할 수 있다. 이들 동작 시퀀스가 완료되면, 휘발성 메모리(120) 내 상태 정보가 도 1과 관련하여 앞서 기재된 시동 시퀀스를 실행한 결과로서 로딩된 것과 비교될 수 있다. 그러나 도 3에 도시된 시퀀스를 이용해 시동 명령어에 응답하기 위해 필요한 시간은 도 1과 관련하여 기재된 시동 시퀀스를 실행하기 위해 필요한 시간보다 짧을 수 있다.

[0061] 도 3에 도시된 예를 들면, 휴면 파일(310)은, 휴면 파일(210)(도 2)과 동일한 포맷을 갖지만, 다른 정보를 포함한다. 덧붙여, 휴면 파일(310)은 휴면 파일(210)과 다른 방식으로 생성된다. 앞서 기재된 것처럼, 휴면 파일(210)(도 2)은 컴퓨팅 장치(100)의 상태를, 휴면 명령어의 시점에서 휘발성 메모리(120)에 나타난 상태로서 기록한다. 이와 달리, 휴면 파일(310)은 셧다운 명령어에 응답하여 생성된다. 그러나 휴면 파일(310)에 포착된 상태 정보가 셧다운 명령어의 시점에서의 컴퓨팅 장치(100)의 완전 상태(full state)를 나타내지 않는다.

[0062] 오히려, 일부 프로세싱은 컴퓨팅 장치(100)를 목표 상태로 두도록 수행될 수 있고, 이때, 휴면 파일(310)이 생성될 수 있다. 도시된 실시예에서, 목표 상태는, 컴퓨팅 장치(100)에의 사용자 로그온은 없는, 운영 체제의 로딩 후 생성될 수 있는 상태를 나타낸다. 이러한 목표 상태는 적어도 부분적으로 셧다운 시퀀스의 일부분을 실행시킴으로써 생성될 수 있다. 예를 들어, 상기 부분은 컴퓨팅 장치(100)의 사용자 또는 사용자들을 로그오프하는 것, 또는 그 밖의 다른 방식으로 사용자 연결을 해제하는 것을 포함할 수 있다. 이러한 프로세싱은 해당 분야에 공지된 바와 같은 기법을 이용해 수행될 수 있다.

[0063] 대안적으로 또는 추가적으로 그 밖의 다른 프로세싱이 컴퓨팅 장치(100)를 목표 상태로 두기 위해 수행될 수 있다. 예를 들어, 프로세싱은 시스템 상태 데이터(148)로부터 더티 데이터(dirty data)를 플러쉬(flushing)하는 것을 포함할 수 있다.

[0064] 덧붙여, 앞서 언급된 것처럼, 셧다운 명령어에 대한 컴퓨팅 장치(100)의 반응의 사용자 기대를 보존하기 위해, 휴면을 포함하는 셧다운 시퀀스가 그 당시 존재할 수 있는 조건을 기초로 하여 조건부로 수행될 수 있다. 마찬가지로 시동 시퀀스는 휴면으로부터의 재개를 조건부로 포함할 수 있다. 도 4, 5 및 6은 이러한 조건부 프로세싱을 도시한다.

[0065] 도 4는 시동 시퀀스, 가령, 시동 명령어에 응답하여 컴퓨팅 장치(100)에 의해 수행될 수 있는 시동 시퀀스를 도시한다. 시동 명령어는, 예를 들어, 사용자가 버튼을 누름으로써, 또는 컴퓨팅 장치(100)로 전력을 공급함으로써, 또는 그 밖의 다른 방식으로 컴퓨팅 장치(100)의 동작을 개시함으로써, 컴퓨팅 장치(100)로 제공될 수 있다.

[0066] 시동 시퀀스가 개시되는 방식에 무관하게, 프로세스는 블록(410)에서 시작될 수 있다. 블록(410)에서, 프로세서(110)는 부트 메모리(154)로부터, 프로세스를 개시하는 명령을 인출(fetch) 및 실행(execute)할 수 있다. 그러나 프로세스의 후기 단계들에서, 명령은 비휘발성 메모리(150)로부터 또는 그 밖의 다른 임의의 적합한 소스로부터, 가령, 네트워크 연결을 통해서, 인출될 수 있다.

[0067] 시동 시퀀스를 개시하기 위해 프로세서(110)를 제어하도록 사용되는 명령의 소스에 무관하게, 프로세스는, 휴면 파일이 비휘발성 메모리(152)에서 검출되는지 여부에 따라 결정 블록(412)에서 분기될 수 있다. 휴면 파일이 비휘발성 메모리에서 검출된 경우, 프로세스는 완료 점 A로 향하여, 도 6에 도시되는 프로세스를 계속할 수 있다. 반대로, 어떠한 휴면 파일도 존재하지 않은 경우, 프로세스는 서브프로세스(450)로 진행될 수 있다.

[0068] 서브프로세스(450)는 해당 분야에 일반적으로 공지된 시동 시퀀스를 구현하는 동작의 시퀀스를 나타낼 수 있다. 이러한 예에서, 블록(420, 422, 424, 426, 428, 430 및 432)에서의 프로세싱은 공지된 시동 시퀀스에서와 같은 프로세싱을 나타낼 수 있다. 그러나 어떠한 적합한 기법을 이용한 어떠한 적합한 동작 시퀀스라도 사용될 수 있

음을 알아야 한다.

[0069] 사용되는 특정 접근법에 무관하게, 서브프로세스(450) 내 프로세싱은 블록(420)에서 시작될 수 있다. 블록(420)에서, 프로세서(110)는 운영 체제 로더(loader)를 실행한다. 이러한 로더는 실행될 때 운영 체제의 구성요소를 비휘발성 메모리(150)로부터 휘발성 메모리(120)로 로딩하는 소프트웨어 구성요소일 수 있다.

[0070] 블록(422)에서, 시스템 상태(140)의 일부로서 생성되는 운영 체제의 이미지를 구성하는 동작이 구성될 수 있다. 이 구성은 임의의 적합한 프로세싱, 가령, 휘발성 메모리로 로딩되는 구성요소의 파라미터의 값을 설정하는 것 또는 시스템 상태(140)의 또 다른 양태를 구성하는 명령을 실행하는 것을 포함할 수 있다.

[0071] 또한 시동 서브프로세스(450)의 일부로서, 컴퓨팅 장치(100)는 장치를 검출할 수 있다. 컴퓨팅 장치(100)로 연결된 임의의 적합한 장치, 가령, 프린터, 네트워크 인터페이스 또는 그 밖의 다른 주변장치가 검출될 수 있다. 검출된 장치를 기초로 하여, 블록(426)에서 드라이버 로더가 실행될 수 있다. 드라이버 로더는 공지된 기법을 이용해 구성되고 드라이버를 로딩하는 소프트웨어 구성요소일 수 있다. 드라이버 로더의 실행은 검출된 장치에 대해 드라이버 소프트웨어를 식별하는 것과 로딩하는 것을 포함할 수 있다. 드라이버가 로딩되면, 이들은 블록(428)에서 시작될 수 있다. 이 프로세싱은 드라이버 및 상기 드라이버가 제어하는 장치를 컴퓨팅 장치(100)로 로딩되는 그 밖의 다른 구성요소에 의한 사용에 이용 가능하게 만들 수 있다.

[0072] 프로세스는 블록(430)으로 진행하며, 여기서 운영 체제 서비스가 시작될 수 있다. 장치 및 운영 체제의 서비스가 사용되기에 이용 가능하면, 프로세싱은 블록(432)으로 진행할 수 있다. 블록(432)에서 애플리케이션 구성요소가 로딩될 수 있다. 이 프로세싱은 해당 분야에서 알려진 것과 같은 기법을 이용해, 또는 그 밖의 다른 임의의 적합한 방식으로, 사용자 로그온 프로세스의 일부로서 수행될 수 있다.

[0073] 애플리케이션 구성요소가 로딩될 때, 도 4에 도시된 프로세스는 블록(432)에서 로딩되는 애플리케이션 구성요소의 속성에 따라 분기될 수 있다. 결정 블록(444)에서의 분기에 의해, 컴퓨팅 장치는 상기 컴퓨팅 장치(100)가 종래의 셧다운 시퀀스가 아닌 휴면을 이용해 셧다운 구성요소를 수행할 경우 하나 이상의 애플리케이션 구성요소가 적절하게 동작하지 않을 때 발생할 수 있는 문제를 개선할 수 있다. 일부 구성요소는 리부트(reboot)를 필요로 할 수 있는데, 이는 컴퓨팅장치가 다음 번에 켜질 때(power on), 완전한 시동 시퀀스가 수행되어, 상태가 로딩 프로세스를 이용해 재-생성되는 것을 의미한다.

[0074] 예를 들어, 도 3에 도시된 휴면을 포함하는 셧다운 시퀀스가 수행되는 경우, 컴퓨팅 장치(100)가 시동하는 시점에 따라 서로 다르게 동작을 수행하는 애플리케이션 구성요소는 사용자가 기대하는 바와 같이 수행하지 않을 수 있다. 이들 구성요소에 대해, 뒤 이은 시동이 수행될 때, 상기 시동이 휴면으로부터의 복원을 기초로 수행되는 경우, 애플리케이션 구성요소는 휴면 파일(310)로부터 복원된 상태 정보를 기초로 구성될 수 있다. 상기 상태 정보는 컴퓨터가 마지막으로 완전한 시동 시퀀스를 수행했을 시점의 표시를 포함할 수 있다. 따라서 상태 정보를 기초로 로딩된 후 구성되는 애플리케이션 구성요소는, 도 4에 도시된 시동 시퀀스가 개시됐을 때를 나타내는 시점 값을 이용해 구성되지 않을 것이다.

[0075] 가능한 사용자 기대와 달리, 상기 구성요소는 완전한 시동 시퀀스가 수행됐던 이전 시점을 나타내는 시점 값에 의해 구성될 것이다. 이 경우, 애플리케이션 구성요소의 거동이 사용자 기대와 다른 시점을 기초로 할 것인데, 사용자가 도 4의 프로세스가 시작한 시점을 기초로 애플리케이션 구성요소가 구성되기를 기대할 것이기 때문이다.

[0076] 따라서 이러한 애플리케이션 구성요소가 컴퓨팅 장치(100)로 로딩될 때, 사용자로부터의 셧다운 명령어에 응답하여 구성요소가 완전한 셧다운 시퀀스를 필요로 함을 결정하는 것이 바람직할 수 있다. 이러한 구성요소가 실행 중일 때, 컴퓨팅 장치는 완전한 셧다운 시퀀스를 수행함으로써 셧다운 명령어에 응답할 수 있다. 이러한 방식으로, 시동 명령어가 다음 번에 수신되면, 어떠한 휴면 파일도 이용 가능하지 않을 것이고, 예를 들어, 도 1에 도시된 바와 같은 완전한 시동 시퀀스가 수행될 것이다. 그 밖의 다른 때에는, 컴퓨팅 장치가 도 3에 도시된 휴면을 포함하는 셧다운 시퀀스를 이용해 셧다운 명령어에 응답할 수 있다.

[0077] 사용자 기대와 일치하는 이러한 거동을 지원하기 위해, 애플리케이션 프로그램이 완전한 셧다운 및 완전한 시동 시퀀스가 수행되기를 필요로 한다고 지정하기 위한 메커니즘이 제공될 수 있다. 도 4의 예에서, 상기 메커니즘은 컴퓨팅 장치(100)의 운영 체제에 의해 지원되는 애플리케이션 프로그래밍 인터페이스(API)를 통해 구현될 수 있다. 모두 경우에서 완전한 셧다운 및 완전한 시동 시퀀스를 필요로 하는 것은 아닌 애플리케이션 구성요소 각각은 이러한 API를 통해 호출할 수 있다.

[0078] 따라서 블록(440)에서 로딩되는 애플리케이션 구성요소가 리부트를 필요로 한다고 결정된 경우, 프로세싱은 블

록(442)으로 분기될 수 있다. 블록(442)에서, 상기 애플리케이션 구성요소를 등록하기 위해 애플리케이션 프로그래밍 인터페이스는 호출될 수 있다. 이러한 예에서, API에 의해, 운영 체제는, 다음 번에 시동 명령어가 수신될 때 리부트를 요청하는 애플리케이션 구성요소가 여전히 실행 중인지 여부를 추적할 수 있다. 그러나 이러한 호출은 어느 때라도 이뤄질 수 있음을 알아야 한다. 가령, 재구성되거나 그 밖의 다른 방식으로 완전 셧다운 및 완전 시동 시퀀스가 수행된다고 결정한 동작 상태와 접하는 임의의 구성요소는 API를 통해 호출할 수 있다.

[0079] 이러한 호출이 API를 통해 이뤄지지 않은 경우, 다음 번에 셧다운 명령어가 수신될 때, 운영 체제는 도 3에 도시된 것과 같은 휴면을 포함하는 셧다운 시퀀스가 사용될 수 있음을 결정할 수 있다. 이와 반대로, 완전 셧다운 및 완전 시동 시퀀스가 요청됐다고 알리는 API를 통해 호출이 이뤄진 경우, 운영 체제는 휴면 파일을 생성하지 않는 완전 셧다운 시퀀스를 수행할 수 있어서, 시동 명령어의 다음 번 수신이 있을 때, 완전한 시동 시퀀스가 수행될 수 있도록 한다.

[0080] 애플리케이션 구성요소가, 완전 셧다운 및 이에 뒤 따른 완전 시동 시퀀스를 포함하는 리부트를 필요로 하는지 여부를 결정하기 위해 임의의 적합한 메커니즘이 사용될 수 있다. 예를 들어, 애플리케이션 구성요소가 블록(442)에서 지시되는 API를 호출하도록 프로그래밍될 수 있다. 대안적으로, 운영 체제는 리부트를 필요로 할 수 있는 기능을 식별하기 위해 로딩될 때 애플리케이션 구성요소를 분석하기 위한 컴퓨터 실행형 명령을 포함할 수 있다. 상기 시나리오에서 결정 블록(440)에서의 프로세싱은 로딩될 때 각각의 애플리케이션 구성요소를 분석하는 것을 포함할 수 있다. 그러나 로딩되는 애플리케이션 구성요소를 기초로 하여 리부트가 필요할 수 있는지 여부를 결정하기 위해 임의의 적합한 기법이 결정 블록(440)에서 사용될 수 있다.

[0081] 도 4는 리부트가 로딩되는 애플리케이션 구성요소를 기초로 요구되는지 여부를 결정하는 것을 도시하지만, 컴퓨팅 장치(100)의 또 다른 요소들에 대해 유사한 프로세싱이 수행될 수 있다. 예를 들어, 운영 체제 구성요소에 대해 유사한 프로세싱이 수행될 수 있다. 대안적으로 또는 추가적으로, 컴퓨팅 장치(100)에 설치되는 장치 또는 상기 컴퓨팅 장치(100)가 연결되는 장치를 기초로 하여 유사한 프로세싱이 수행될 수 있다.

[0082] 리부트에 대한 필요성을 표시할 수 있는, 결정 블록(440)에서 식별되는 조건(condition)에 무관하게, 이들 조건이 식별되는 경우, 프로세싱은 블록(442)으로 분기되며, 여기서 상기 표시가 저장된다. 상기 표시는 사용자로부터의 셧다운 명령어에 응답하여 완전 셧다운을 트리거, 또는 이에 추가로, 또는 이를 대신하여, 휴면 파일이 이용 가능한 경우라도, 시동하기 위한 사용자 명령어에 응답하여 완전 시동 시퀀스를 트리거할 수 있다. 이들 조건이 검출되지 않은 경우, 프로세싱은 블록(444)으로 진행할 수 있다.

[0083] 블록(444)에서, 컴퓨팅 장치(100)가 휴면을 포함하는 시동 시퀀스를 이용하는 것의 유효성(effectiveness)을 결정할 수 있도록 하는 데이터가 수집될 수 있다. 이 예에서, 블록(444)에서의 프로세싱이, 이 예에서 완전 시동 시퀀스의 실행을 표시하는 서브프로세스(450)를 수행한 시점을 기록한다. 이 정보는 임의의 적합한 방식으로 기록될 수 있다. 예를 들어, 시동 시점에 대한 정보는 비휘발성 메모리(152)에 기록될 수 있다. 상기 정보는 개별 시동 시점으로서 기록될 수 있으며, 이는 완전 시동 시퀀스가 수행된 시점마다 완전 시동 시퀀스를 수행하도록 요구되는 시점을 표시한다. 대안적으로, 정보는 복수의 완전 시동 시퀀스의 이동 평균(running average)으로서 또는 그 밖의 다른 적합한 방식으로 기록될 수 있다.

[0084] 시동 시점에 대한 정보는 블록(444)에서 임의의 적합한 방식으로 결정될 수 있다. 하나의 예를 들면, 타이머(timer)가 서브프로세스(450)의 개시에서 시작되고, 프로세싱이 블록(444)에 도달할 때 읽힐 수 있다. 그러나 그 밖의 다른 시 측정 기법이 알려져 있으며, 블록(444)에서 적용될 수 있다.

[0085] 시동 시점이 기록되면, 프로세싱은 블록(446)으로 진행할 수 있다. 이때, 컴퓨팅 장치(100)의 종래의 동작이 발생할 수 있다. 이러한 동작은 셧다운 명령어가 수신될 때까지 계속될 수 있다.

[0086] 도 5는 이러한 셧다운 명령어에 응답하여 수행될 수 있는 프로세싱을 도시한다. 도 5에 도시된 프로세스는 해당 분야에서 알려진 것과 같은 기법을 이용하는 컴퓨팅 장치(100)의 동작을 나타내는 블록(510)을 포함한다. 동작 동안, 셧다운 명령어(512)가 수신될 수 있다. 셧다운 명령어(512)는 사용자 입력에 의해 임의의 적합한 방식으로, 가령, 그래픽 사용자 인터페이스 또는 하드웨어 제어를 통해, 생성될 수 있다.

[0087] 일부 실시예에서, 컴퓨팅 장치(100)는 셧다운 시퀀스를 트리거할 수 있는 복수 유형의 사용자 입력을 지원할 수 있다. 도 7은 그래픽 사용자 인터페이스를 도시하며, 상기 그래픽 사용자 인터페이스를 통해 사용자는 셧다운 명령어를 입력할 수 있다. 이 예에서, 컴퓨터 운영 체제에 의해 제공되는 사용자 인터페이스 상에 나타나는 "시작"이라고 라벨링된 버튼을 누름으로써, 그래픽 사용자 인터페이스(710)가 불러와 진다(invoke). 그러나 서로 다른 운영 체제가 서로 다른 인터페이스를 지원하고 사용자 인터페이스를 불러오기 위한 임의의 적합한 기법이

사용될 수 있음을 알아야 한다.

[0088] 상기 버튼의 누름에 응답하여, 해당 분야의 공지된 기법을 이용해, 그래픽 사용자 인터페이스(710)가 운영 체제에 의해 제공될 수 있다. 그래픽 사용자 인터페이스(710)를 통해, 컴퓨팅 장치(100)의 사용자는 복수의 가능한 명령어들 중에서 컴퓨팅 장치 상에서의 현재 세션을 종료시키기 위한 것을 선택할 수 있다. 이때 3개의 옵션이 나타난다.

[0089] 명령어(714)는 여기서 "셧다운"이라고 라벨링된다. 이러한 셧다운 명령어는 많은 컴퓨팅 장치에서 통상적인 것이며, 전통적으로 컴퓨팅 장치가 완전 셧다운 시퀀스를 수행해야 함을 표시하도록 사용되었다. 그러나 도 5에 도시된 실시예에서, 사용자가 셧다운 명령어(714)를 선택함으로써, 컴퓨팅 장치(100)의 운영 체제가 휴면을 포함하는 부분 셧다운 시퀀스가 대신 수행될 수 있는지 여부를 결정하게 된다. 이 실시예에서, 운영 체제는 사용자에게 시맨틱 의미(semantic meaning)를 갖는 명령어에 대한 라벨을 상기 의미와 불일치할 가능성이 있는 방식으로 이용한다. 그럼에도, 조건부 프로세싱이 사용자 기대를 보존한다.

[0090] 그러나 사용자가 완전 셧다운 시퀀스가 수행됨을 보장하기를 원하는 경우, 이러한 이유로, 서로 다른 라벨을 갖는 개별 명령어가 제공될 수 있다. 사용자가, 다음 번 시동 명령어가 있을 때, 비휘발성 메모리(150)로부터 소프트웨어를 로딩하고, 비휘발성 메모리(152)로부터의 데이터를 이용해 이를 구성함으로써, 운영 체제 상태가 생성되도록, 휴면 파일을 생성하지 않고 컴퓨팅 장치에게 완전 셧다운을 수행하라고 지시하기를 원하는 경우, 사용자는 명령어(715)를 선택할 수 있다. 이 예에서, 명령어(715)는 "리부트(reboot)"라고 라벨링된다. 이러한 라벨링은 사용자에게, 다음 번 시동 명령어가 있을 때, 완전 시동 시퀀스가 수행되도록 완전 셧다운 시퀀스가 수행될 것임을 알리기 위해 사용된다. 이 경우, 명령어(715)는 "셧다운" 명령어가 발행될 때 종래의 컴퓨팅 시스템에서 수행되는 동작과 유사한 동작을 수행한다. 그러나 그래픽 사용자 인터페이스(710)를 제공하는 컴퓨팅 장치에서, 전통적인 셧다운 명령어와 연관된 시맨틱 라벨이 명령어(714)로 적용되었다. 따라서 명령어(715)에는 다른 라벨이 부여된다.

[0091] 그래픽 사용자 인터페이스(710)는 또한 사용자 세션을 종료하기 위한 또 다른 옵션을 포함할 수 있다. 이 예에서, 그래픽 사용자 인터페이스(710)는 명령어(716)를 포함한다. 명령어(716)가 선택되면, 컴퓨팅 장치(100)가 지정된 사용자에 대한 세션을 해제함으로써 응답할 수 있다. 컴퓨팅 장치의 이러한 거동은 해당 분야에 공지되어 있다. 이 경우, 명령어(716)는 종래의 로그오프 명령어에 대응할 수 있다. 많은 적합한 명령어 옵션이 그래픽 사용자 인터페이스(710)에 포함될 수 있지만, 도시된 실시예에서는 명령어(714 또는 716)의 선택만 도 5에 도시된 프로세스의 개시를 야기한다.

[0092] 셧다운 명령어가 수신되는 방식 및 명령어의 속성에 무관하게, 명령어의 수신에 응답하여, 프로세싱이 블록(510)에서 블록(514)으로 전환될 수 있다. 블록(514)에서, 셧다운 시퀀스의 일부분을 시작하는 것이 수행될 수 있다. 블록(514)에서 수행되는 셧다운 시퀀스의 일부분은 종래의 프로세싱을 포함할 수 있다. 이 예에서, 블록(514)에서의 프로세싱이 컴퓨팅 장치(100) 상에서의 임의의 사용자 세션(들)을 종료한다. 도 1과 관련하여 앞서 기재된 바와 같이, 이러한 프로세싱은 애플리케이션을 종료하고 사용자 상태 데이터(134)를 저장하거나 그 밖의 다른 임의의 적합한 동작을 수행하는 것을 포함할 수 이다. 이들 동작의 결과로서, 하나의 사용자 세션에서 다음 번 사용자 세션까지 영속(persist)되는 사용자 상태(130) 내 임의의 정보가 사용자 상태 데이터(134)로부터 비휘발성 메모리, 가령, 비휘발성 메모리(152)로 이동된다.

[0093] 사용자 세션 또는 그 밖의 다른 영속 사용자 상태 데이터(134)를 종료하기 위해 취해지는 특정 단계에 무관하게, 이들 단계가 완료될 때, 프로세싱은 결정 블록(516)으로 진행될 수 있다. 결정 블록(516)에서, 도 5의 프로세스는 리부트가 요청됐는지 여부에 따라 분기될 수 있다. 블록(516)에서의 프로세싱은 임의의 적합한 방식으로 수행될 수 있다. 결정 블록(516)에서 리부트가 요청됐는지 여부를 결정하기 위해 임의의 하나 이상의 기준이 적용될 수 있다. 하나의 예를 들면, 리부트가 요청됐는지 여부를 결정하기 위해 블록(516)에서 사용자 입력이 사용될 수 있다. 예를 들어, 사용자가 리부트 명령어(715)를 선택할 때(도 7), 상기 사용자 선택은 리부트가 요청됐다는 표시로서 기능할 수 있다.

[0094] 또 다른 예를 들면, 도 4와 관련하여, 가령, 블록(442)(도 4)에서 API를 호출함으로써, 애플리케이션 구성요소가 리부트를 요청할 수 있다고 기재되었다. 이러한 호출이 이뤄진 경우, 결정 블록(516)에서의 프로세싱은 리부트가 요청되었다는 것을 판정할 수 있다. 일부 실시예에서지만, 결정 블록(516)에서의 프로세싱은 복수의 기준에 따라 조절될 수 있다. 예를 들어, 프로세싱은 블록(442)에서 애플리케이션 구성요소가 API로의 호출을 통해 리부트에 대한 요청을 등록했다고 결정할 수 있다. 결정 블록(516)에서의 추가 프로세싱이 이러한 요청이 시행(honor)되어야 하는지 여부를 결정할 수 있다. 이러한 프로세싱은, 예를 들어, 요청하는 애플리케이션 구성요소

가 도 5의 프로세스가 실행되는 시점에서 여전히 실행 중인지 여부를 결정하는 것을 포함할 수 있다. 대안적으로 또는 추가적으로, 결정 블록(516)에서의 프로세싱은, 리부트를 명령내리기 위해, 요청하는 구성요소가 충분한 액세스 권한(access privilege)을 갖는지 여부를 결정하는 것을 수반할 수 있다.

[0095] 결정 블록(516)에서 수행되는 프로세싱의 속성에 무관하게, 상기 프로세싱의 결과로서, 리부트가 요청됐다고 결정되면, 프로세스는 블록(530)으로 분기된다. 이 시나리오에서, 블록(530)은 완전 셧다운 시퀀스를 나타낸다. 이러한 완전 셧다운 시퀀스는 해당 분야에 공지된 바와 같이 수행될 수 있다. 이는 사용자 세션을 해제하는 것, 더티 데이터를 플러쉬(flush)하는 것, 및 컴퓨팅 장치로의 전력을 차단하는 것(power off)을 수반할 수 있다. 셧다운 시퀀스를 수행하는 데 취해지는 특정 단계에 무관하게, 완료되면, 도 5의 프로세스가 종료되어, 컴퓨팅 장치(100)는 전력 차단 상태로 유지될 수 있다.

[0096] 역으로, 결정 블록(516)에서 리부트가 요청되지 않은 경우, 프로세스는 결정 블록(518)으로 진행할 수 있다. 결정 블록(518)에서의 프로세싱은, 완전 셧다운 시퀀스가 수행되어야 하는지, 또는 부분 셧다운이 수행되고 그 후 휴면이 수행되어야 하는지를 결정하기 위한 조건부 프로세싱의 일례이다. 일반적으로 결정 블록(518)에서의 프로세싱은 임의의 적합한 정책(policy)의 적용을 수반할 수 있다. 이러한 정책은 셧다운 명령어가 수신된 시점에서 평가(evaluate)될 수 있다.

[0097] 도시된 예에서, 적용되는 정책은 휴면을 이용할 때 얻어지는 시간 절약과 관련된다. 결정 블록(518)에서, 휴면으로부터의 시동에 의해 시간 절약이 이뤄지는지 여부가 결정될 수 있다. 이러한 결정은, 컴퓨팅 장치(100)를 완전 시동 시퀀스 또는 휴면으로부터의 재개 및 이에 뒤 따르는 부분 시동 시퀀스를 포함하는 동작 상태로 두는 상대적 시간들에 대한 기록된 정보를 비교함으로써, 이뤄질 수 있다. 완전 시동을 수행하기 위한 시점에 대한 정보는, 예를 들어, 블록(444)(도 4)에서 저장된 정보를 기초로 할 수 있다. 컴퓨팅 장치(100)를 휴면으로부터의 재개 후 동작 상태로 두기 위해 필요한 시간에 대한 정보가 도 6의 프로세스의 실행의 종료 시 기록된 정보를 기초로 하는 것과 유사한 방식으로 판단될 수 있다.

[0098] 재개 및 이에 뒤 따르는 부분 시동을 기초로 하는 동작 상태를 생성하기 위한 시간이, 완전 시동을 수행하기 위한 시간보다 길 경우, 프로세싱은 결정 블록(518)으로부터 서브프로세스(530)로 분기될 수 있다. 반대로, 결정 블록(518)에서의 프로세싱이 휴면으로부터의 재개 및 이에 뒤 따르는 시동 시퀀스의 부분 실행이 바람직하다고 결정한 경우, 프로세싱은 결정 블록(520)으로 진행할 수 있다.

[0099] 결정 블록(520)에서, 컴퓨팅 장치(100)가 휴면을 포함하는 부분 셧다운 시퀀스에 대해 적절한 상태인지 여부를 결정하기 위한 추가 조건부 프로세싱이 수행될 수 있다. 이러한 프로세싱은, 현재 세션 동안, 임의의 구성요소에 대해 구성 변화가 특정되었는지 여부를 결정하는 것을 수반할 수 있다. 이러한 구성 변화가 리부트가 효과적이 될 것을 요구하는 경우, 휴면을 포함하는 셧다운은 컴퓨팅 장치(100)의 거동에 대한 사용자 기대를 이행하지 않을 수 있는데, 왜냐하면 셧다운 명령어(714)(도 7)를 선택하는 것이 전통적으로 컴퓨팅 장치가 다음 번 시동에서 구성 변화를 적용하게 할 라벨과 연관되기 때문이다.

[0100] 컴퓨팅 장치(100)가 다음 번 시동이 있을 때 전통적으로 완전 시동을 표시하도록 사용되는 라벨을 갖는 명령어에 응답하여 휴면을 포함한 셧다운 시퀀스를 이행한 경우, 이들 구성요소의 상태는 구성 변화를 기초로 하는 상태 대신 그들의 이전 상태를 재개할 것이다. 따라서 다른 경우라면 완전 셧다운 시퀀스와 연관됐을 명령어를 불러오는 사용자 기대가 기대한 거동을 보이지 않을 시나리오가 존재할 수 있다. 컴퓨팅 장치(100)가 기대되는 사용자 거동에 불일치하는 식으로 동작하는 것을 피하기 위해, 도 5의 프로세스는, 사용자 기대와 일치하는 동작을 획득하기 위해 컴퓨팅 장치가 완전 셧다운 시퀀스가 수행되어야 함을 자동으로 결정하는지 여부에 따라 분기될 수 있다. 그럴 경우, 프로세스는 서브프로세스(530)로 분기될 수 있으며, 여기서 완전 셧다운 시퀀스가 앞서 설명된대로 수행된다.

[0101] 도시된 실시예에서, 완전 셧다운 시퀀스가 수행되어야 할 조건은, 임의의 구성요소가 현재 세션 중에 구성 설정(configuration settings)을 변경했는지 여부를 결정함으로써, 식별된다. 이 결정을 하기 위한 기법은 해당 분야에 공지되어 잇고 결정 블록(520)에서 적용될 수 있다. 하나의 예를 들면, 구성요소의 실행의 구성 설정을 변경하기 위한 프로세싱은 플래그(flag)를 설정하거나 그 밖의 다른 방식으로 구성 변화의 표시를 기록하는 것을 수반할 수 있다. 상기 시나리오에서, 결정 블록(520)에서의 프로세싱은 상태 플래그의 값을 체크하는 것을 수반할 수 있다. 그러나 이에 추가로, 또는 이를 대체하여, 적합한 프로세싱이 사용될 수 있다. 예를 들어, 프로세싱은 적용되지 않은(unapplied) 구성 설정을 검출하기 위해 하나 이상의 메모리 위치를 스캐닝하는 것을 수반할 수 있다.

[0102] 결정 블록(520)에서의 결정이 이뤄지는 방식에 무관하게, 완전 셧다운 및/또는 뒤 이은 완전 시동이 요구될 어떠한 조건도 존재하지 않을 경우, 프로세싱은 결정 블록(522)으로 진행할 수 있다. 블록(522)에서, 컴퓨팅 장치(100)를 휴면이 발생되는 목표 상태로 완전히 두기 위한 동작이 수행된다. 앞서 도 3과 관련하여 설명된 바와 같이, 상기 목표 상태는 운영 체제 상태가 유지되지만 모든 사용자 세션은 해제되며, 사용자의 다음 번 로그온 시 요구되는 임의의 사용자 상태가, 적절한 형태로 비휘발성 메모리에 영속(persist)되는 상태에 대응할 수 있다.

[0103] 이 목표 상태를 얻기 위해 수행될 수 있는 동작의 일례가 더티 데이터(dirty data)의 플러쉬(flush)이다. 이에 추가로 또는 이를 대체하여, 시스템 상태 데이터(148)의 일부로서 저장되는 그 밖의 다른 데이터가 로그온된 사용자의 세션과 관련된 경우, 블록(522)에서의 프로세싱은 데이터를 비휘발성 메모리(152)에 저장하는 것을 수반할 수 있다.

[0104] 목표 상태를 완전히 이루기 위해 수행되는 어떤 동작이 수행되는지에 무관하게, 그 후, 프로세싱은 블록(524)으로 진행할 수 있다. 블록(524)에서, 컴퓨팅 장치(100) 상에 목표 상태를 재생성하기 위한 휴면 파일의 적합성(suitability)을 확신하기 위해 추후 사용될 수 있는 정보가 취해질 수 있다. 예를 들어, 일부 컴퓨팅 장치는 복수의 운영 체제 또는 하나의 운영 체제의 복수의 인스턴스(instance)를 이용해 구성될 수 있다. 한 운영 체제의 특정 인스턴스의 셧다운의 일부로서 생성된 휴면 파일이 사용되어, 상기 운영 체제의 상기 인스턴스를 시동하기 위한 명령어에 응답해서만 운영 체제 상태를 복원할 수 있다.

[0105] 그러나 휴면 파일이 생성될 때 사용된 것이 아닌 운영 체제를 이용해 컴퓨팅 장치가 동작될 수 있는 가능성이, 휴면 파일이 생성된 시점과 상기 파일을 기초로 상태의 재-생성을 트리거할 뒤 이은 시동 명령어 사이에 컴퓨팅 장치 상에서 운영 체제가 실행될 가능성을 생성한다. 또 다른 운영 체제 또는 동일한 운영 체제의 인스턴스에 의한 중간 동작(intervening operation)이 휴면 파일에 포착된 상태가 사용자 기대에 일치하는 동작을 얻기 위해 생성되어야 할 컴퓨팅 장치의 상태를 더 이상 나타내지 않을 가능성을 만들 수 있다.

[0106] 예를 들어, 사용자가, 제 1 운영 체제에 의한 셧다운 동작 후, 제 2 운영 체제를 로딩하고 제 1 운영 체제에 의해 사용되는 임의의 데이터 또는 그 밖의 다른 구성요소를 변경한 경우, 이 인스턴스에서의 휴면 파일로부터의 재개에 의해, 중간 사용자 변경을 반영하지 않은 상태의 생성을 야기할 것이다.

[0107] 따라서 다음 번 시동 명령어가 있은 후 휴면 파일이 컴퓨팅 장치(100)의 동작 상태를 재생성하는 데 사용되기 적합하지 여부를 결정하기 위한 메커니즘이 채용될 수 있다. 도 5에 도시된 실시예에서, 상기 메커니즘은 휴면 파일이 생성된 시점에서의 정보를 저장하는 것을 수반한다. 이 특정 예에서, 상기 정보는 파일 시스템에 의해 유지되는 시퀀스 번호이다. 특히, 시퀀스 번호는 NTFS 파일 시스템 또는 컴퓨팅 장치에서 동작할 수 있는 그 밖의 다른 파일 시스템에 의해 유지될 수 있다. 이러한 시퀀스 번호는 디스크 저장소의 공간(volume)이 로딩될 때마다 증분(increment)될 수 있다. 따라서 블록(524)에서의 프로세싱은 휴면 파일 및 운영 체제와 연관된 그 밖의 다른 데이터를 포함하는 공간과 연관된 NTFS 시퀀스 번호를 저장하는 것을 수반할 수 있다. 다음 번 시동 명령어와 관련하여 액세스될 수 있도록 이 값은 비휘발성 메모리에 저장될 수 있다.

[0108] 휴면 파일의 사용성(usability)의 추후 결정을 가능하게 하기 위해 블록(524)에서 기록되는 특정 정보와 무관하게, 프로세스는 서브프로세스(526)로 진행할 수 있다. 상기 서브프로세스(526)는 휴면 파일을 저장하는 것을 포함할 수 있다. 블록(526)에서의 프로세싱은 컴퓨팅 장치의 휴면과 연관된 종래의 기법을 이용해 수행될 수 있다. 그러나 휴면 파일을 수행하는 임의의 적합한 기법이 사용될 수 있음을 알아야 한다.

[0109] 서브프로세스(526)의 일부로서 휴면 파일을 저장하기 위해 사용되는 특정 기법에 무관하게, 휴면 파일을 저장하면, 컴퓨팅 장치(100)로의 전력이 차단될 수 있다. 시동 명령어가 수신될 때까지 컴퓨팅 장치(100)는 파워 다운(power down)된 채 유지될 수 있다.

[0110] 도 4 및 도 6에서 도시되는 바와 같이 다음 번 시동 명령어가 프로세싱될 수 있다. 도 4는 시작 명령어를 수신하는 것에 응답하여 수행될 수 있는 프로세싱을 도시한다. 상기 프로세싱은 블록(410) 및 결정 블록(412)에서의 프로세싱을 포함할 수 있다. 프로세싱이 결정 블록(412)에 도달할 때, 상기 프로세스는, 휴면 파일이 존재하는지 여부에 따라, 분기될 수 있는데, 가령, 바로 이전 셧다운 동안 서브프로세스(526)가 수행된 경우 휴면 파일이 존재할 수 있다. 상기 휴면 파일이 존재하는 경우, 도 4의 프로세스는 A라고 라벨링된 커넥터를 통해 도 6에 도시된 것과 같이 프로세싱을 지속하는 것으로 분기될 수 있다.

[0111] 도 6에서의 프로세싱은 휴면 파일이 존재할 때 수행될 수 있는 프로세스를 도시한다. 도 6의 프로세스는 블록(601)에서 시작할 수 있다. 상기 블록(601)에서, 프로세스는 블록(412)(도 4)에서 검출된 휴면 파일이 셧다운

동안 목표 상태를 포착한 휴면 파일, 가령, 서브프로세스(526)과 관련하여 표시되는 휴면 파일을 나타내는지 여부에 따라 분기될 수 있다. 그럴 경우, 프로세스는 결정 블록(610)으로 진행할 수 있다.

[0112] 대안적으로, 휴면 파일이 시스템 상태 정보에 추가로 사용자 상태를 포함하는 종래의 휴면 파일을 나타낼 수 있다. 이러한 휴면 파일은 상기 상태를 복원하기 위한 종래의 기법에 따라 사용될 수 있다. 종래의 프로세싱은 서브프로세스(670)에서 수행될 수 있고, 여기서 휴면 파일은 이전 휴면 시점에서의 컴퓨팅 장치의 상태, 가령, 사용자 상태를 재확립하기 위해 사용될 수 있다. 서브프로세스(670)의 완료 후, 프로세싱은 블록(638)으로 진행할 수 있다.

[0113] 반대로, 결정 블록(601)에서 결정된 휴면 파일이 셧다운 프로세스의 일부로서 기록된 경우, 프로세스는 결정 블록(610)으로 계속된다. 결정 블록(610)에서 시작하여, 시동 명령어에 응답하여, 전체 시동 시퀀스가 수행되거나 휴면으로부터의 재개 및 이에 뒤 따른 부분 시동 시퀀스가 수행되어야 하는지 여부를 결정하기 위해 하나 이상의 동작이 수행될 수 있다. 이 예에서, 휴면 파일이 존재한다고 결정된 때라도 휴면으로부터의 재개가 수행되어야 하는지 여부를 결정하기 위해 복수의 조건이 체크된다. 결정 블록(610)에서 체크되는 이러한 한 가지 조건은 하드웨어 구성의 변화가 있어서, 컴퓨팅 장치(100)에 대해 휴면으로부터의 재개가 현재 컴퓨터 구성과 일치하지 않는 상태 정보를 재-생성하는 것을 야기할 수 있는지 여부를 결정하는 것을 수반한다. 이러한 변화는 임의의 적합한 방식으로, 가령, 컴퓨팅 장치의 마지막 세션 동안 생성되고 비휘발성 메모리에 저장된 하드웨어 구성요소의 인벤토리(inventory)를 체크함으로써 검출될 수 있다. 인벤토리의 각각의 아이템이 설치됨을 보장하기 위해 다음 번 시동이 있을 때 컴퓨팅 장치의 상기 하드웨어 구성이 체크될 수 있다. 그러나 인벤토리를 체크하는 것은 이러한 프로세싱이 수행될 수 있는 방식의 하나의 예에 불과함을 알아야 한다. 결정이 이뤄지는 방식에 무관하게, 하드웨어 구성이 변경된 경우, 프로세싱은 결정 블록(610)으로부터 서브프로세스(650)로 분기될 수 있다. 서브프로세스(650)는 운영 체제를 재로딩하는 것을 수반할 수 있다. 서브프로세스(650)에서의 프로세싱은 해당 분야에 공지된 바와 같은 기술을 이용해 수행될 수 있다. 서브프로세스(650)에서의 운영 체제의 로딩 후에, 프로세스는 블록(632)으로 진행할 수 있다.

[0114] 이와 달리, 결정 블록(610)에서의 프로세싱이 어떠한 하드웨어 구성도 발생하지 않았다고 결정한 경우, 프로세싱은 블록(612)으로 진행할 수 있다. 블록(612)에서, 컴퓨팅 장치(100)가 휴면으로부터의 재개가 수행될 상태에 있는지 여부를 동적으로 결정하도록 추가 프로세싱이 수행될 수 있다. 이 경우, 결정 블록(612)에서의 프로세싱은 블록(524)(도 5)에서 저장된 정보를 사용하여, 휴면으로부터의 재개가 수행된 경우 휴면 파일의 생성 사이에 변경이 발생하여 사용자 기대가 충족되지 않을지 여부를 결정한다.

[0115] 이 예에서, 블록(612)에서의 프로세싱은 휴면 파일을 포함하는 공간(volume)과 연관된 NTFS 시퀀스 번호를 체크하는 것을 포함한다. 휴면 파일이 생성되었기 때문에 상기 공간에 로딩되지 않은 경우, 블록(612)에서 판독된 시퀀스 번호는, 시동 시의 시퀀스 번호의 가능성을 나타내는 알려진 크기(amount)만큼, 블록(524)에서 저장된 시퀀스 번호와 다를 것이다. 반대로, 시퀀스 번호의 차이가 알려진 크기보다 큰 경우, 블록(612)에서의 프로세싱은 휴면 파일의 생성 시점과 휴면으로부터의 재개를 트리거한 시동 명령어 시점 사이에 변화가 있었을 가능성이 있다고 알려줄 것이다.

[0116] 결정 블록(620)에서, 프로세스는 블록(612)에서 수행된 비교를 기초로 분기될 수 있다. 시퀀스 번호가 일치하지 않을 경우, 프롤세스는 서브프로세스(750)로 분기한다. 이러한 분기는 시퀀스 번호의 차이가 휴면 파일이 사용자 기대와 일치하는 컴퓨팅 장치의 동작 상태를 확립하지 않았을 수 있음을 나타낼 때 발생할 수 있다. 따라서 운영 체제 소프트웨어를 재로딩함으로써 시스템 상태(140)가 생성되는 서브프로세스(650)가 수행된다.

[0117] 이와 달리, 블록(612)에서 수행되는 비교가 시퀀스 번호가 일치한다고 가리키는 경우, 프로세스는 서브프로세스(630)로 진행할 수 있다. 분기가 취해질 때, 휴면 파일은 컴퓨팅 장치의 상태를 재확립하기에 적절하다고 결정되었다. 따라서 서브프로세스(630)는 휴면 파일로부터 컴퓨팅 장치의 목표 상태를 재확립하는 것을 수반한다. 서브프로세스(630)는 휴면으로부터의 재개를 위한 공지된 기술을 이용해여 수행될 수 있다. 그러나 이 시나리오에서, 사용자 상태를 포함하는 컴퓨팅 장치의 상태를 재확립하는 대신 휴면 파일을 기초로 재개하는 것이 휴면 파일이 생성된 시점에서의 목표 상태를 재-생성한다. 예를 들어 이 상태는 서브프로세스(526)(도 5)의 시작에서의 컴퓨팅 장치의 상태를 나타낼 수 있다. 그러나 그 밖의 다른 실시예에서, 부분 사용자 상태가 휴면 파일에 저장될 수 있는데, 가령, 이는 운영 체제가 시동 시퀀스의 완료 후 사용자에 의해 애플리케이션이 열리고(open), 상기 애플리케이션이 여전히 열려 있는 동안의 컴퓨팅 장치의 상태를 포착(capture)하도록 휴면 파일을 저장할 가능성이 높다고 예측할 때 발생할 수 있다.

[0118] 서브프로세스(630)가 완료되면, 도 6의 프로세스는 블록(632)으로 진행할 수 있다. 프로세싱이 서브프로세스

(630 또는 650)를 통해 블록(632)에 도달하는지 여부에 무관하게, 블록(632)에서 시동 명령어에 응답하기로 요청된 시점이 기록될 수 있다. 기록되는 값의 의미는 프로세싱이 블록(632)에 도달하는 경로에 따라 달라질 수 있다. 프로세싱이 서브프로세스(630)를 통해 블록(632)에 도달할 때, 상기 시점은 프로세싱의 일부로서 휴면으로부터의 재개를 이용한 시동 시점을 나타내고, 이에 따라 기록된다. 반대로, 상기 프로세싱이 서브프로세스(650)를 통해 블록(632)에 도달할 때, 상기 시점은 완전 시동 시퀀스를 이용한 시동 시점을 나타내고, 이에 따라 기록된다. 블록(632)에서의 프로세싱은 임의의 적합한 방식으로, 가령, 블록(444)(도 4)과 관련하여 기재된 것과 같은 깁버을 이용하여, 수행될 수 있다.

[0119] 블록(632)에서의 정보 기록의 결과로, 결정 블록(518)(도 5)에서의 프로세싱은 휴면으로부터의 재개와 시동 시퀀스의 일부분을 포함하는 완전 시동 시퀀스를 기초로 하는 시동 명령어에 응답할 시점을 비교하기 위해 이용 가능한 정보를 가질 수 있다. 이 정보는 기록되고 임의의 적합한 방식으로 비교될 수 있다.

[0120] 그 후 프로세싱은 서브프로세스(634)로 진행될 수 있다. 서브프로세스(634)에서 시동 시퀀스의 일부분은 컴퓨팅 장치에 대한 바람직한 동작 상태를 생성하도록 수행될 수 있다. 이 일부분은 사용자 로그온을 포함할 수 있다. 이 동작은 공지된 방식으로 수행될 수 있고, 자동 로그온을 포함하거나, 사용자가 로그온 프로세스를 수동으로 수행하기 위한 정보를 제공할 수 있는 로그온 스크린을 제공하는 것을 포함할 수 있다. 프로세싱이 서브프로세스(650)를 통해 서브프로세스(634)에 도달하는 시나리오에서, 서브프로세스(650)와 서브프로세스(634)에서의 프로세싱의 조합이 완전 시동 시퀀스를 나타낼 수 있다. 이와 달리, 프로세싱이 서브프로세스(630)를 통해 서브프로세스(634)에 도달한 경우, 시동 명령어에 대한 응답은 휴면으로부터의 재개와 시동 시퀀스의 일부분을 포함한다.

[0121] 이 예에서, 시동 시퀀스의 일부분이 서브프로세스(634)에서 사용자 로그온을 나타낸다. 이러한 프로세싱은 종래의 기법을 이용해 수행될 수 있다. 그러나 휴면으로부터의 재개 후의 시동 시퀀스를 완료하기 위해 사용되는 특정 단계들이 임의의 적합한 기법일 수 있다.

[0122] 그 후 상기 프로세스는 블록(638)으로 진행할 수 있고, 여기서 상기 휴면 파일은 무효화(invalidate)될 수 있다. 또한 프로세싱은 서브프로세스(670) 후 블록(638)에 도달할 수 있다. 프로세싱이 블록(638)에 도달하는 방식에 무관하게, 휴면 파일은 올바르지 않은 동작 상태(incorrect operating state)를 재-생성할 수 있을 때 상기 휴면 파일이 추후 사용되지 않도록 하는 임의의 적합한 방식으로 무효화될 수 있다. 상기 휴면 파일은, 예를 들어, 그 내용을 임의의 방식으로 변경함으로써, 또는 파일이 무효가 되는 별도 메모리 구조에 기록함으로써, 또는 상기 파일을 삭제함으로써, 무효화될 수 있다.

[0123] 그 후 도 6의 프로세스는 종료될 수 있다. 프로세스가 종료될 때, 컴퓨팅 장치(100)는 동작 상태이도록 구성될 수 있고, 그 후 셧다운 또는 리부트 명령어가 수신될 때까지 계속 동작할 수 있다.

[0124] 도 8은 본 발명이 구현될 수 있는 적합한 컴퓨팅 시스템 환경(800)의 하나의 예를 도시한다. 상기 컴퓨팅 시스템 환경(800)은 적합한 컴퓨팅 환경의 하나의 예에 불과하며, 본 발명의 사용 또는 기능의 범위에 어떠한 제약도 한정하는 것을 의도하지 않는다. 컴퓨팅 환경(800)은 예시적 동작 환경(800)에서 도시된 구성요소들 중 임의의 하나 또는 이들의 조합과 관련된 어떠한 종속성 또는 요건도 갖는 것으로 해석되어서는 안 된다.

[0125] 본 발명은 그 밖의 다른 많은 범용 또는 특수 목적의 컴퓨팅 시스템 환경 또는 구성과 함께 동작된다. 본 발명과 함께 사용되기에 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경, 및/또는 구성의 예로는, 개인용 컴퓨터, 서버 컴퓨터, 핸드-헬드, 또는 랩톱 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 시스템, 셋 톱 박스, 프로그램 가능한 소비 전자기기, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 상기의 시스템 또는 장치 중 임의의 것을 포함하는 분산 컴퓨팅 환경, 및 등이 있지만, 이에 제한되지는 않는다.

[0126] 상기 컴퓨팅 환경은 컴퓨터-실행가능한 명령, 가령, 프로그램 모듈을 실행할 수 있다. 일반적으로 프로그램 모듈은 특정 작업을 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 객체, 구성요소, 데이터 구조 등을 포함한다. 본 발명은 또한 작업이 통신 네트워크를 통해 링크된 원격 프로세싱 장치에 의해 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 메모리 저장 장치를 포함하는 로컬 컴퓨터 저장 매체와 원격 컴퓨터 저장 매체 모두에 위치할 수 있다.

[0127] 도 8을 참조하면, 본 발명을 구현하기 위한 예시적 시스템이 컴퓨터(810)의 형태로 된 범용 컴퓨팅 장치를 포함한다. 컴퓨터(810)의 구성요소는, 프로세싱 유닛(820), 시스템 메모리(830), 및 시스템 메모리를 포함하는 다양한 시스템 구성요소들을 프로세싱 유닛(820)으로 연결하는 시스템 버스(821)를 포함할 수 있지만, 이에 제한되지는 않는다. 상기 시스템 버스(821)는 몇 가지 유형의 버스 구조 중 임의의 것일 수 있으며, 가령, 메모리 버

스 또는 메모리 제어기, 주변 장치 버스, 및 다양한 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스일 수 있다. 비-제한적 예를 들면, 이러한 아키텍처는 산업 표준 아키텍처(ISA: Industry Standard Architecture) 버스, 마이크로 채널 아키텍처(MCA: Micro Channel Architecture) 버스, 강화된 ISA(EISA: Enhanced ISA) 버스, 비디오 일렉트로닉스 표준 연합(VESA: Video Electronics Standards Association) 로컬 버스, 및 메자닌 버스(Mezzanine bus)라고도 알려진 주변 장치 인터커넥트(PCI: Peripheral Component Interconnect) 버스를 포함한다.

[0128] 일반적으로 컴퓨터(810)는 다양한 컴퓨터 판독형 매체를 포함한다. 컴퓨터 판독형 매체는 컴퓨터(810)에 의해 액세스될 수 있는 임의의 이용 가능한 매체일 수 있으며, 휘발성 매체와 비휘발성 매체, 이동식 및 비-이동식 매체를 모두 포함한다. 비-제한적 예를 들면, 컴퓨터 판독형 매체는 컴퓨터 저장 매체, 및 통신 매체를 포함할 수 있다. 컴퓨터 저장 매체는 정보, 가령, 컴퓨터 판독형 명령, 데이터 구조, 프로그램 모듈, 또는 그 밖의 다른 데이터의 저장을 위해 임의의 방법 또는 기법으로 구현되는 휘발성과 비휘발성, 이동식 및 비-이동식 매체를 모두 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 그 밖의 다른 메모리 기법, CD-ROM, 디지털 다용도 디스크(DVD), 또는 그 밖의 다른 광학 디스크 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장장치 또는 그 밖의 다른 자기 저장 장치, 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨터(810)에 의해 액세스될 수 있는 그 밖의 다른 임의의 매체를 포함하지만, 이에 제한되지 않는다. 일반적으로 통신 매체는 변조된 데이터 신호, 가령, 반송파 또는 그 밖의 다른 전송 메커니즘으로 된 컴퓨터 판독형 명령, 데이터 구조, 프로그램 모듈, 또는 기타 데이터를 저장하며 임의의 정보 전달 매체를 포함한다. "변조된 데이터 신호"라는 용어는 자신의 특성 중 하나 이상이 신호 내 정보를 인코딩하기 위한 방식으로 설정 또는 변경된 신호를 의미한다. 비제한적 예를 들면, 통신 매체는 유선 매체, 가령, 유선 네트워크 또는 직접 배선 연결과, 무선 매체, 가령, 음향, RF, 적외선 및 그 밖의 다른 무선 매체를 포함한다. 상기 중 임의의 것의 조합이 또한 컴퓨터 판독형 매체의 범위 내에 포함되어야 한다.

[0129] 시스템 메모리(830)는 휘발성 및/또는 비휘발성 메모리의 형태로 된 컴퓨터 저장 매체, 가령, 리드 온리 메모리(ROM)(831) 및 랜덤 액세스 메모리(RAM)(832)를 포함한다. 가령, 일반적으로, 시동(startup) 동안, 컴퓨터(810) 내 요소들 간 정보를 전송하는 데 도움이 되는 기본 루틴을 포함하는 기본 입/출력 시스템(833)(BIOS)이 ROM(831)에 저장된다. 일반적으로 RAM(832)은 프로세싱 유닛(820)에 의해 즉시 액세스 가능한, 및/또는 현재 작동 중인 데이터 및/또는 프로그램 모듈을 포함한다. 비-제한적 예를 들면, 도 8은 운영 체제(834), 애플리케이션 프로그램(835), 기타 프로그램 모듈(836), 및 프로그램 데이터(837)를 도시한다.

[0130] 컴퓨터(810)는 그 밖의 다른 이동식/비-이동식 휘발성/비휘발성 컴퓨터 저장 매체를 더 포함할 수 있다. 단지 예로서, 도 8은 비-이동식 비휘발성 자기 매체로부터 읽거나 쓰는 하드 디스크 드라이브(840), 이동식 비휘발성 자기 디스크(852)로부터 읽거나 쓰는 자기 디스크 드라이브(851), 및 이동식 비휘발성 광학 디스크(856), 가령, CD ROM 또는 그 밖의 다른 광학 매체로부터 읽거나 쓰는 광학 디스크 드라이브(855)를 도시한다. 하드 디스크 드라이브(840)는 자기 매체로부터 읽거나 쓸 수 있는 하나 이상의 자기 헤드를 갖는 스피닝 자기 매체(spinning magnetic medium)로서 구현될 수 있다. 예시적 동작 환경에서 사용될 수 있는 그 밖의 다른 이동식/비-이동식, 휘발성/비휘발성 컴퓨터 저장 매체로는, 자기 테이프 카세트, 플래시 메모리 카드, 디지털 다목적 디스크, 디지털 비디오 테이프, 솔리드 스테이트 RAM, 솔리드 스테이트 ROM, 및 기타 등등이 포함되며, 이에 제한되지 않는다. 하드 디스크 드라이브(841)는 비-이동식 메모리 인터페이스, 가령 인터페이스(840)를 통해 시스템 버스(821)로 연결되며, 자기 디스크 드라이브(851) 및 광학 디스크 드라이브(855)는 이동식 메모리 인터페이스, 가령 인터페이스(850)에 의해 시스템으로 연결되는 것이 일반적이다.

[0131] 도 8에서 도시되고 앞서 설명된 드라이브 및 이들의 연관된 컴퓨터 저장 매체는 컴퓨터 판독형 명령, 데이터 구조, 프로그램 모듈 및 그 밖의 다른 컴퓨터(810)용 데이터의 저장을 제공한다. 도 8에서, 예를 들어, 하드 디스크 드라이브(841)는 운영 체제(844), 애플리케이션 프로그램(845), 기타 프로그램 모듈(846), 및 프로그램 데이터(847)를 저장하는 것으로 도시된다. 이들 구성요소는 운영 체제(834), 애플리케이션 프로그램(835), 기타 프로그램 모듈(836), 및 프로그램 데이터(837)와 동일하거나 다를 수 있다. 운영 체제(844), 애플리케이션 프로그램(845), 기타 프로그램 모듈(846), 및 프로그램 데이터(847)은 최소한 서로 다른 카피임을 설명하기 위해 이들에게 서로 다른 번호가 부여된다. 사용자는 입력 장치, 가령 키보드(862), 및 일반적으로 마우스, 트랙볼 또는 터치 패드라고 일컬어지는 위치 지시 장치(pointing device)(861)를 통해 명령어 및 정보를 컴퓨터(810)로 입력할 수 있다. 그 밖의 다른 입력 장치(도면에 도시되지 않음)는 마이크로폰, 조이스틱, 게임 패드, 위성 접시, 스캐너, 등을 포함할 수 있다. 이러한 그리고 그 밖의 다른 입력 장치는, 시스템 버스로 연결되지만 그 밖의 다른 인터페이스 및 버스 구조로는 연결되지 않을 수 있는 사용자 입력 인터페이스(860), 가령, 병렬 포트, 게임

포트, 또는 전역 직렬 버스(USB)를 통해 프로세싱 유닛(820)으로 연결된다. 모니터(891) 또는 그 밖의 다른 유형의 디스플레이 장치가 인터페이스, 가령, 비디오 인터페이스(890)를 통해 시스템(821)으로 연결된다. 모니터에 추가로, 컴퓨터는 그 밖의 다른 주변 출력 장치, 가령, 스피커(897) 및 프린터(896)를 더 포함할 수 있으며, 이들은 출력 주변 인터페이스(895)를 통해 연결될 수 있다.

[0132] 컴퓨터(810)는 하나 이상의 원격 컴퓨터, 가령, 원격 컴퓨터(880)로의 논리 연결을 이용해 네트워크 연결된 환경에서 동작할 수 있다. 상기 원격 컴퓨터(880)는 개인용 컴퓨터, 서버, 라우터, 네트워크 PC, 피어 장치(peer device) 또는 그 밖의 다른 일반적인 네트워크 노드일 수 있으며, 도 8에 메모리 저장 장치(881)만 도시되었어도, 일반적으로 컴퓨터(810)와 관련해 앞서 기재된 요소들 중 다수 또는 모두를 포함한다. 도 8에 도시된 논리적 연결은 로컬 영역 네트워크(LAN)(871) 및 광역 네트워크(WAN)(873)를 포함하지만, 그 밖의 다른 네트워크도 포함할 수 있다. 이러한 네트워킹 환경은 사무실에서 일반적이며, 기업별 컴퓨터 네트워크 인트라넷 및 인터넷이다.

[0133] LAN 네트워킹 환경에서 사용될 때, 컴퓨터(810)는 네트워크 인터페이스 또는 어댑터(870)를 통해 LAN(871)으로 연결된다. WAN 네트워킹 환경에서 사용될 때, 일반적으로 컴퓨터(810)는 모뎀(872) 또는 WAN(873), 가령 인터넷을 통한 통신을 확립하기 위한 그 밖의 다른 수단을 포함한다. 내부형이거나 외부형일 수 있는 모뎀(872)은 사용자 입력 인터페이스(860) 또는 그 밖의 다른 적절한 메커니즘을 통해 시스템 버스(821)로 연결될 수 있다. 네트워크 연결된 환경에서, 컴퓨터(810) 또는 이의 일부분과 관련되어 설명되는 프로그램 모듈은 원격 메모리 저장 장치에 저장될 수 있다. 비-제한적 예를 들면, 도 8은 원격 애플리케이션 프로그램(885)을 메모리 장치(881) 상에 상수하는 것으로 도시한다. 도시된 네트워크 연결은 예시이고, 컴퓨터들 간 통신 링크를 확립하기 위한 그 밖의 다른 수단이 사용될 수 있다.

[0134] 따라서 본 발명의 적어도 하나의 실시예의 몇 가지 양태가 기재됐지만, 해당 분야의 통상의 기술자에게 다양한 변형, 수정, 및 개선이 자명할 것이다. 예를 들어, 완전 시동 시퀀스를 수행할지 또는 휴면으로부터의 재개 및 이에 뒤 따른 시동 시퀀스의 일부를 수행할지에 대한 결정이 각각의 시퀀스를 수행하기 위해 관측되는 상대적 시간을 기초로 이뤄진다. 셧다운에서 유사한 프로세싱이 수행될 수 있음을 알아야 한다. 셧다운에서 수행되는 경우, 결정은 휴면 파일을 저장하는지 또는 저장하지 않는지에 의해 구현될 수 있다. 따라서 시동 시 발생하는 것으로 기재된 동작은 셧다운 시에도 수행될 수 있고, 그 반대도 가능함을 알아야 한다.

[0135] 앞서 기재된 이점은 그 밖의 다른 방식으로도 이뤄질 수 있다. 예를 들어, 상태 설정 프로세스 중에 컴퓨터의 CPU 및 그 밖의 다른 구성요소, 가령, 디스크에 의한 일을 피하기 위해, 이러한 접근법에 의해 셧다운 명령어에 응답하여 데이터가 휴면 파일에 저장될 수 있고, 이는 다음 번 시동 명령어에 대한 응답 동안 및/또는 시동 명령어가 완료된 후, 사용자의 경험을 빠르게 하는 데 도움이 될 것이다. 예를 들어, 사용자가 로그온할 때, 복수의 애플리케이션(가령, WINDOWS EXPLORER 웹 브라우저, 시동 앱(apps) 등이 런칭될 수 있다. 운영 체제는 시동 명령어의 프로세싱이 완료된 후 지정된 간격 동안 사용자가 액세스하는 파일(및 이들의 오프셋)을 명시적으로 추적할 수 있다. 이들 애플리케이션 또는 그 밖의 다른 구성요소가 메모리로부터 읽혀서 셧다운 명령어의 다음 번 프로세싱 동안 생성될 휴면 파일로 저장될 수 있다. 이러한 방식으로, 이들 애플리케이션 또는 그 밖의 다른 구성요소가, 이들 애플리케이션을 런칭하는 것의 일부로서 랜덤하게 읽어야 할 필요 없이, 디스크로부터 메모리로 순차적으로 읽힐 것이다.

[0136] 또한 사용자 로그온 및 로그오프가 기재된다. 셧다운 명령어는 복수의 사용자가 하나의 컴퓨터에 로그온하는 시나리오에서 제공될 수 있다. 셧다운 시퀀스가 부분적으로 수행되고, 그 후, 휴면 동작이 수행되는 경우, 부분 셧다운 시퀀스는 복수의 사용자의 로그오프를 야기할 수 있지만, 그럼에도, 앞서 기재된 기법이 적용될 수 있다.

[0137] 예를 들어, 앞서 기재된 기법은 사용자 개입 없는 자동화된 서비스(automated servicing)를 제공하도록 사용될 수 있다. 예를 들어, 부분 셧다운 시퀀스를 수행하고 그 후 휴면을 수행함으로써 셧다운 명령어에 응답한 컴퓨팅 장치가 사용자 활동이 기대되지 않을 때, 가령, 한밤 중에, 자동으로 깨도록(wake) 구성될 수 있다. 깨어나면, 컴퓨팅 장치는 유지관리 활동(maintenance activity), 가령, 소프트웨어 업데이트 제공을 수행할 수 있다. 유지관리 활동이 사용자에게 투명하도록, 이는 사용자에게, 컴퓨팅 장치가 하루의 끝에 셧다운되는 것처럼 보인다. 이러한 능력은, 예를 들어 컴퓨팅 장치가, 셧다운 명령어에 응답하여, 자신이 유지관리 활동 또는 적용 및 보호를 위한 패치(patch)를 갖고 있다고 검출하여 적절한 시점에 깨어나는 경우, 구현될 수 있다. 컴퓨팅 장치가 깰 때, 상기 컴퓨팅 장치는 어떠한 필요한 유지관리 장치라도, 가령, 패치 적용을 수행한다. 그 후 시스템은 완전 재시작을 하고 그 후 다시 부분 셧다운 및 이에 뒤따르는 휴면을 수행한다. 이 시나리오에 의해 소프트웨

어 제조사는 유지관리 활동을 사용자에게 안보이도록 하는 솔루션을 제공할 수 있다. 이 능력은 소비자와 기업 PC 모두에 적용될 수 있다. 사용자 경험을 개선하는 것에 추가로, 이러한 접근법은 또한 전력, 특히, 기업 사용자를 위한 전력을 절약해 줄 수 있다.

[0138] 이러한 변경, 변형 및 개선은 본 명세서의 일부이며, 본 발명의 사상 및 범주 내에 속한다. 따라서, 전술한 설명 및 도면은 단지 일 예일 뿐이다.

[0139] 본 발명의 앞서 기재된 실시예는 임의의 다양한 방식으로 구현될 수 있다. 예를 들어, 실시예는 하드웨어, 소프트웨어, 또는 이들의 조합을 이용해 구현될 수 있다. 소프트웨어로 구현될 때, 단일 컴퓨터에서 제공되거나 복수의 컴퓨터에서 분산되거나 관계 없이, 소프트웨어 코드는 임의의 적합한 프로세서 또는 프로세서들의 집합에서 실행될 수 있다. 이러한 프로세서는 집적 회로로서 구현될 수 있으며, 여기서 집적 회로 구성요소 내 하나 이상의 프로세서를 가진다. 그러나 프로세서는 임의의 적합한 형식으로 회로를 이용해 구현될 수 있다.

[0140] 덧붙여, 컴퓨터는 많은 형태 중 임의의 것, 가령, 선반-장착형 컴퓨터(rack-mounted computer), 데스크톱 컴퓨터, 랩톱 컴퓨터, 또는 태블릿 컴퓨터로 구현될 수 있다. 추가로, 컴퓨터는 일반적으로 컴퓨터로 간주되지 않지만 적합한 프로세싱 능력을 갖는 장치, 가령, PDA, 스마트 폰 또는 그 밖의 다른 임의의 적합한 휴대용 또는 고정형 전자 장치에 이식될 수 있다.

[0141] 또한 컴퓨터는 하나 이상의 입력과 출력 장치를 가질 수 있다. 이들 장치는 다른 것들 중에서, 사용자 인터페이스를 제공하도록 사용될 수 있다. 사용자 인터페이스를 제공하기 위해 사용될 수 있는 출력 장치의 예로는, 프린터 또는 출력의 비주얼 표시를 위한 디스플레이 스크린 및 스피커 또는 그 밖의 다른 출력의 가청 표시를 위한 소리 발생 장치가 있을 수 있다. 사용자 인터페이스를 위해 사용될 수 있는 입력 장치의 예로는 키보드, 및 위치 지시 장치, 가령, 마우스, 터치 패드, 및 디지털화된 태블릿이 있다. 또 다른 예로서, 컴퓨터는 음성 인식 또는 그 밖의 다른 가청 포맷을 통해 입력 정보를 수신할 수 있다.

[0142] 이러한 컴퓨터는 임의의 적합한 형태의 하나 이상의 네트워크, 가령, 로컬 영역 네트워크 또는 광역 네트워크, 가령, 기업 네트워크 또는 인터넷에 의해 상호연결될 수 있다. 이러한 네트워크는 임의의 적합한 기법을 기초로 할 수 있고, 임의의 적합한 프로토콜에 따라 동작할 수 있으며, 무선 네트워크, 유선 네트워크 또는 광섬유 네트워크를 포함할 수 있다.

[0143] 또한 본원에서 제시되는 다양한 방법 또는 프로세스가 다양한 운영 체제 또는 플랫폼 중 임의의 하나를 채용하는 하나 이상의 프로세서에 의해 실행될 수 있는 소프트웨어로 코딩될 수 있다. 덧붙여, 이러한 소프트웨어는 많은 적합한 프로그래밍 언어 및/또는 프로그래밍 또는 스크립트 툴 중 임의의 것을 이용해 써질 수 있고, 또한 프레임워크나 가상 머신에서 실행되는 실행 가능한 기계 언어 코드 또는 중간 코드로서 컴파일될 수 있다.

[0144] 이와 관련해, 본 발명은 컴퓨터 판독형 저장 매체(또는 복수의 컴퓨터 판독형 매체)(가령, 컴퓨터 메모리, 하나 이상의 플로피 디스크, 컴팩트 디스크(CD), 광학 디스크, 디지털 비디오 디스크(DVD), 자기 테이프, 플래시 메모리, 현장 프로그램 가능한 게이트 어레이(Field Programmable Gate Array) 내 회로 구성 또는 그 밖의 다른 반도체 장치 또는 그 밖의 다른 비일시적(non-transitory) 유형(tangible)의 컴퓨팅 저장 매체)로서 구현될 수 있고 하나 이상의 컴퓨터 또는 또 다른 프로세서 상에서 실행될 때 앞서 언급된 본 발명의 다양한 실시예를 구현하는 방법을 수행하는 하나 이상의 프로그램에 의해 인코딩될 수 있다. 앞서 기재된 바와 같은 본 발명의 다양한 양태를 구현하기 위해 저장된 프로그램이 하나 이상의 서로 다른 컴퓨터 또는 그 밖의 다른 프로세서로 로딩될 수 있도록 컴퓨터 판독형 저장 매체는 수송형일 수 있다. 본원에서 사용될 때, 용어 "비-일시적 컴퓨터 판독형 저장 매체"는 제조될 것으로 고려될 수 있는 컴퓨터 판독형 매체(즉, 제조 물품) 또는 기계만 포함한다. 이에 추가로, 또는 이를 대체하여, 본 발명은 컴퓨터 판독형 저장 매체가 아닌 다른 컴퓨터 판독형 매체, 가령, 전파 신호로서 구현될 수 있다.

[0145] 용어 "프로그램" 또는 "소프트웨어"는 본원에서, 앞서 언급된 것과 같은 본 발명의 다양한 양태를 구현하기 위해 컴퓨터 또는 그 밖의 다른 프로세서를 프로그램하도록 사용될 수 있는 임의의 유형의 컴퓨터 코드 또는 컴퓨터 실행 가능형 명령의 세트를 지칭하는 일반적인 의미로 사용된다. 덧붙여, 이 실시예의 하나의 양태에 따르면, 실행될 때 본 발명의 방법을 수행하는 하나 이상의 컴퓨터 프로그램은 단일 컴퓨터 또는 프로세서 상에 상주할 필요는 없으며, 복수의 서로 다른 컴퓨터 또는 프로세서에 걸쳐 모듈식으로 분산되어, 본 발명의 다양한 양태를 구현할 수 있다.

[0146] 컴퓨터 실행형 명령은 많은 형태, 가령, 하나 이상의 컴퓨터 또는 그 밖의 다른 장치에 의해 실행되는 프로그램 모듈로 존재할 수 있다. 일반적으로 프로그램 모듈은 특정 작업을 수행하거나, 특정 추상 데이터 유형을 구현하

는 루틴, 프로그램, 객체, 구성요소, 데이터 구조, 등을 포함한다. 일반적으로 프로그램 모듈의 기능은 다양한 실시예에서 원하는 대로 조합 또는 분산될 수 있다.

[0147] 또한 데이터 구조는 컴퓨터 판독형 매체에 임의의 적합한 형태로 저장될 수 있다. 설명의 간결성을 위해, 데이터 구조는 데이터 구조 내 위치를 통해 관련 필드를 갖는 것으로 나타날 수 있다. 이러한 관계는 필드들 간 관계를 담는 컴퓨터 판독형 매체 내 위치를 갖는 필드에 대해 저장공간을 할당함으로써 이뤄질 수 있다. 그러나 임의의 적합한 메커니즘이 데이터 구조의 필드들 내 정보들 간 관계를 확립하는 데 사용될 수 있으며, 가령, 포인터, 태그, 또는 데이터 요소들 간 관계를 확립하는 그 밖의 다른 메커니즘을 사용하는 것이 있다.

[0148] 본 발명의 다양한 양태는 홀로 또는 조합되어 또는 상기에서 기재된 실시예에서 특정하게 언급되지 않은 다양한 배열로 사용될 수 있으며, 따라서 상기의 기재 또는 도면의 도시에서 제공되는 구성요소들의 세부사항 및 배열에 국한되지 않는다. 예를 들어, 하나의 실시예에 기재된 양태는 또 다른 실시예에 기재된 양태와 임의의 방식으로 조합될 수 있다.

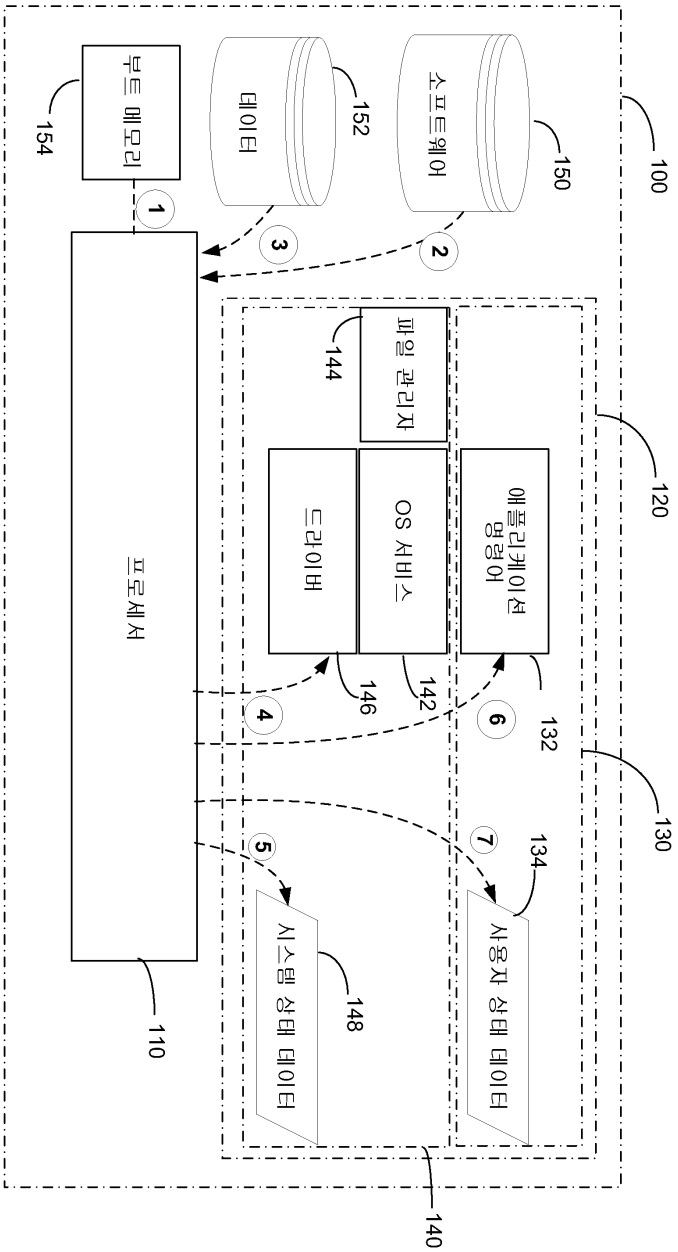
[0149] 또한 본 발명은 방법으로 구현될 수 있으며, 이의 예가 제공되었다. 방법의 일부로서 수행되는 동작들은 임의의 적합한 방식으로 순서화될 수 있다. 따라서 동작들이 설명된 순서와 다른 순서로 수행되는 실시예가 구성될 수 있는데, 가령, 예시적 실시예에서 순차적인 동작으로 나타나더라도 일부 동작들은 동시에 수행될 수도 있다.

[0150] 청구항에서 청구항 요소를 수정하기 위한 "제1의", "제2의", "제3의" 등 같은 서수 용어의 사용은 그 자체로는 하나의 청구항 요소에 대하 다른 청구항 요소의 어떠한 우선순위, 선행, 또는 순서 또는 방법의 동작들이 수행되는 시간적 순서도 내포하지 않으며, 특정 명칭을 갖는 하나의 청구항 요소를 동일한 명칭을 갖는 또 다른 청구항 요소와 구별하기 위한 라벨로서 사용(서수 용어로서 사용)될 뿐이다.

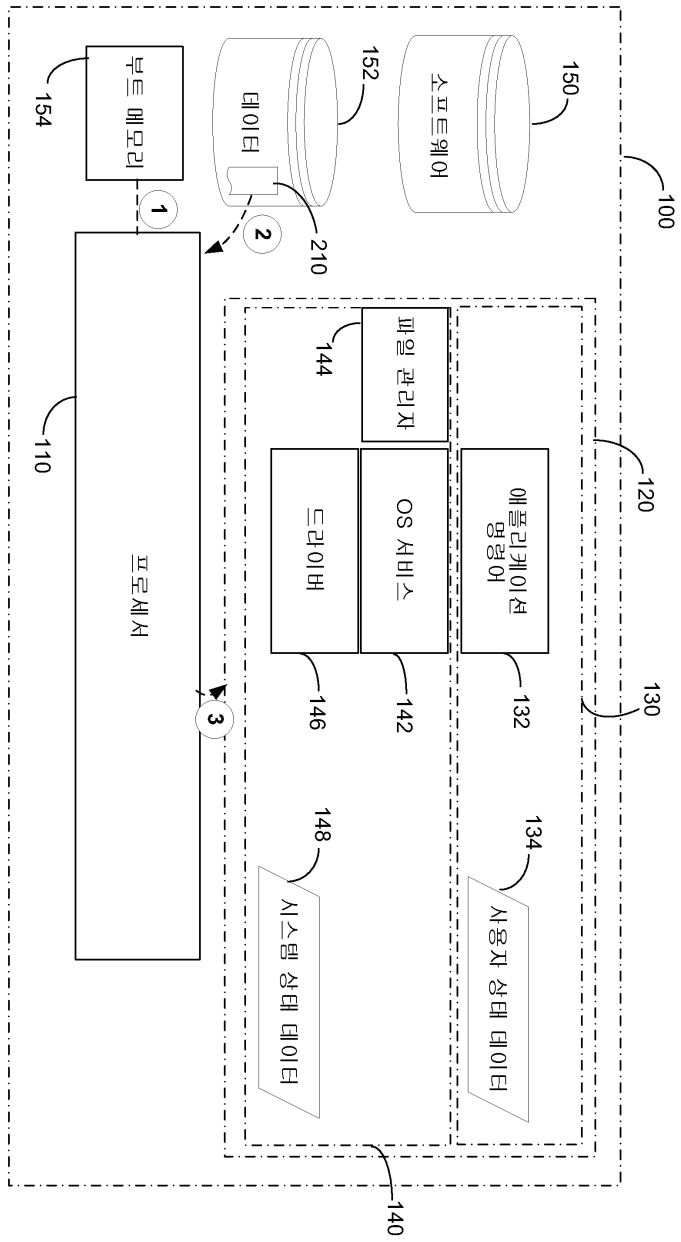
[0151] 또한, 본원에서 사용되는 구문과 단어는 설명을 위한 것이며 한정을 위한 것이 아니다. "를 포함하는(including)", "를 포함하는(comprising)" 또는 "를 갖는(having)", "를 내포하는(containing)", "를 포함하는(involving)" 및 이들의 변형은 이들 앞에 나열되는 아이템들 및 이의 균등물뿐 아니라 추가 아이템까지 포함하는 것을 의미한다.

도면

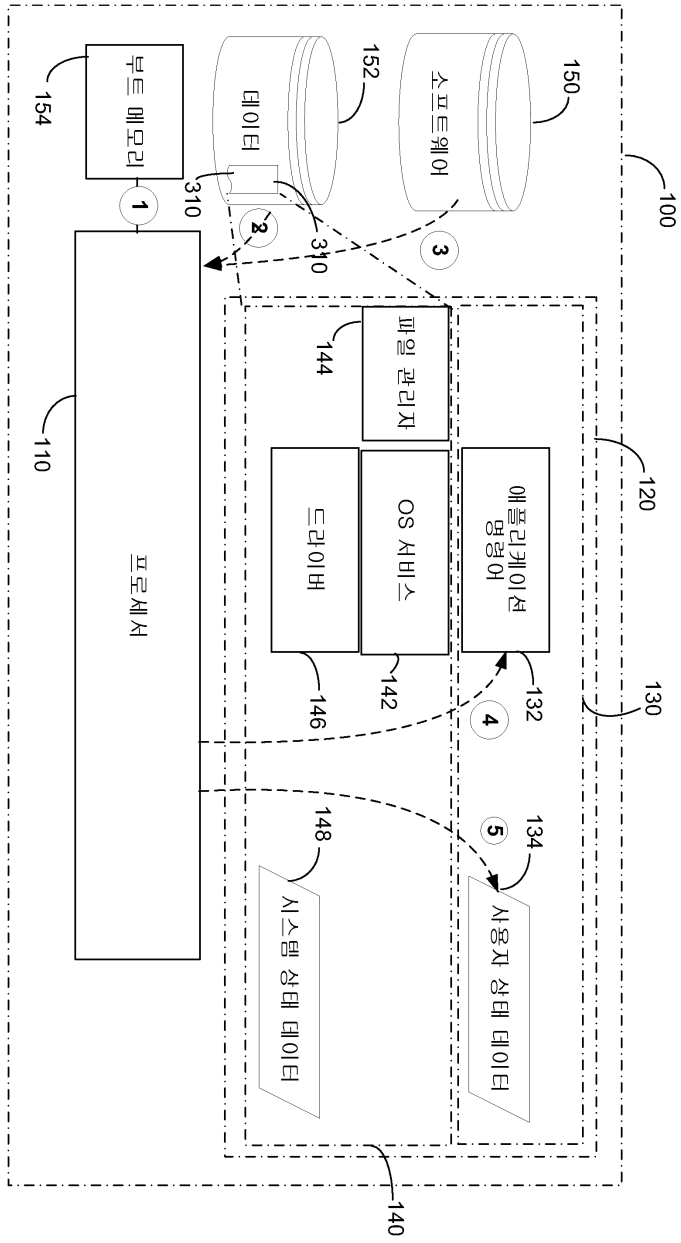
도면1


100
부트 메모리
154
데이터
152
소프트웨어
150
1
2
3
프로세서
110
120
파일 관리자
144
드라이버
OS 서비스
애플리케이션 엔진어
130
132
142
146
4
5
6
7
시스템 상태 데이터
148
사용자 상태 데이터
134
140

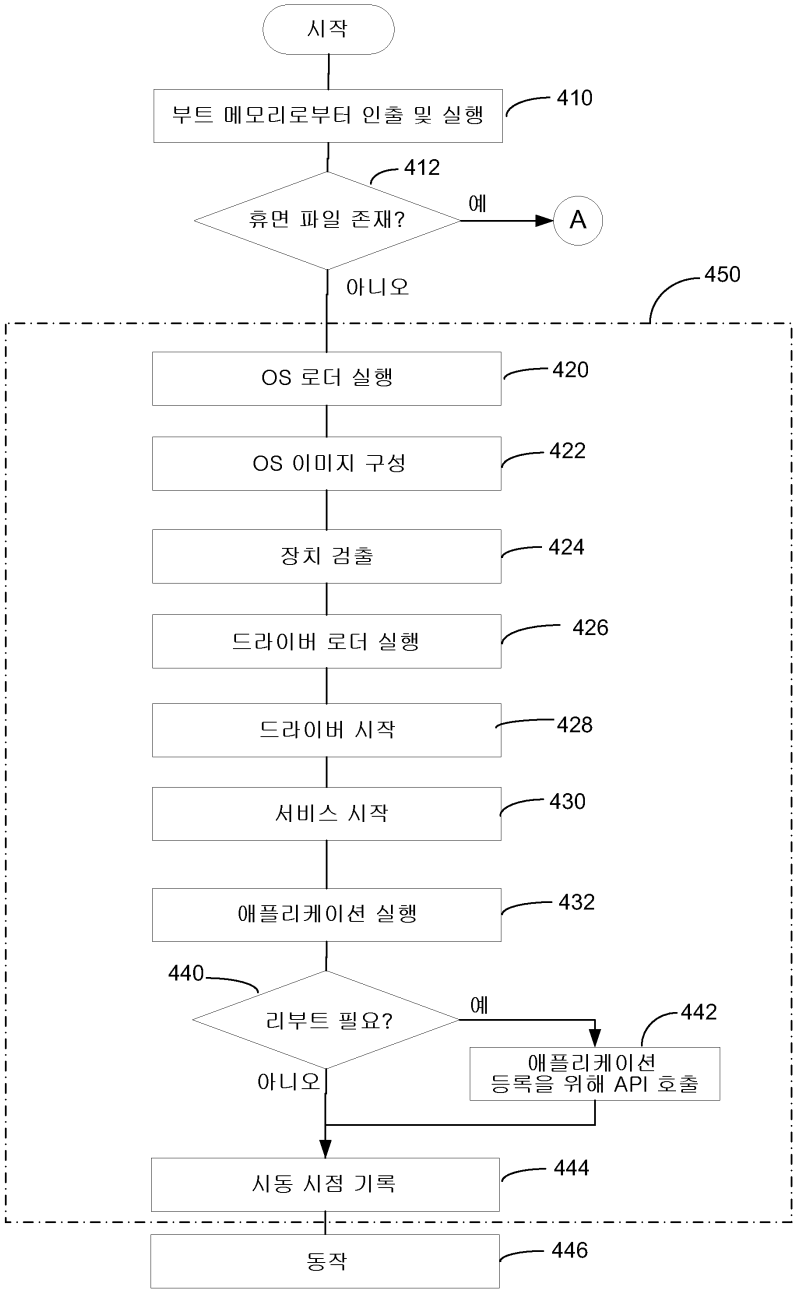
도면2


100
150
소프트웨어
152
데이터
210
154
부트 메모리
1
2
110
프로세서
3
120
130
140
144
파일 관리자
146
드라이버
142
OS 서비스
132
애플리케이션 명령어
148
시스템 상태 데이터
134
사용자 상태 데이터

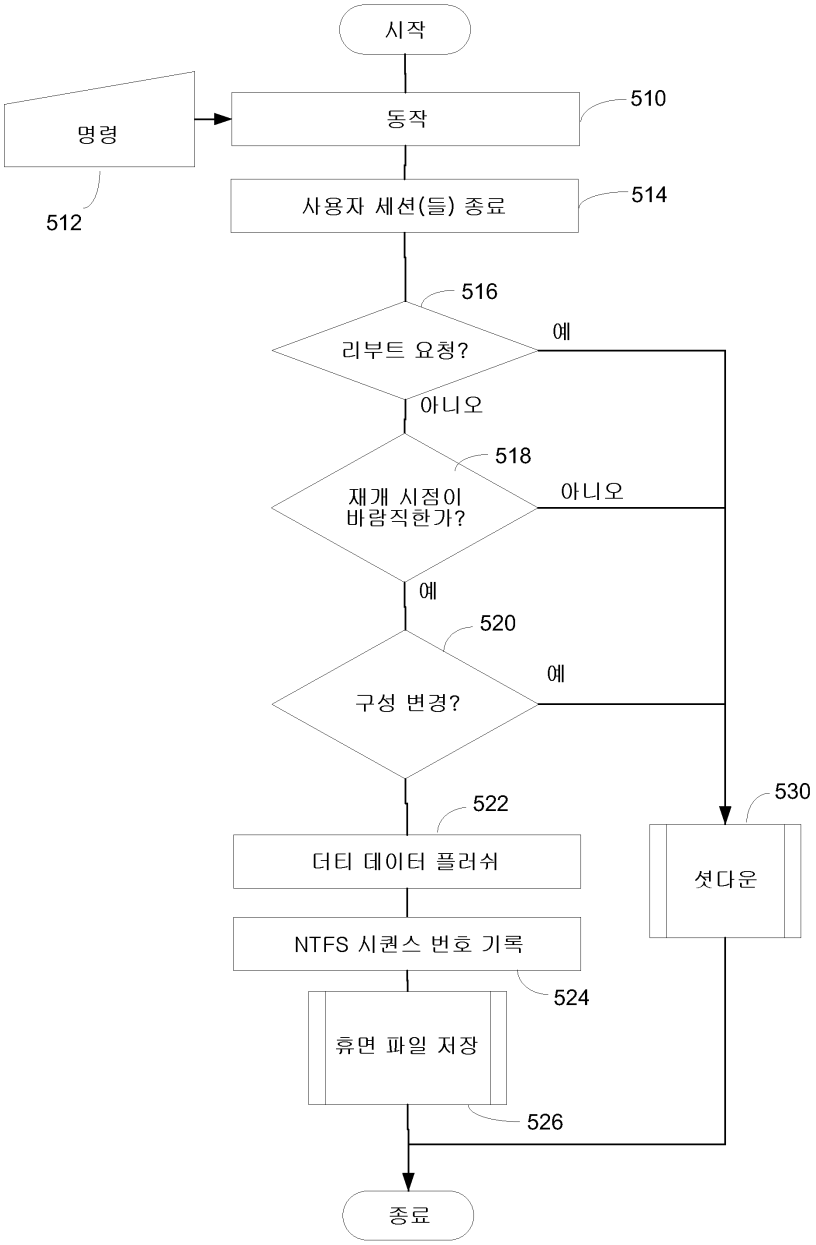
도면3


100
부트 메모리
154
데이터
152
소프트웨어
150
310
310
1
2
3
110
프로세서
120
파일 관리자
144
드라이버
146
OS 서비스
142
애플리케이션 명령어
132
130
4
5
134
148
시스템 상태 데이터
사용자 상태 데이터
140

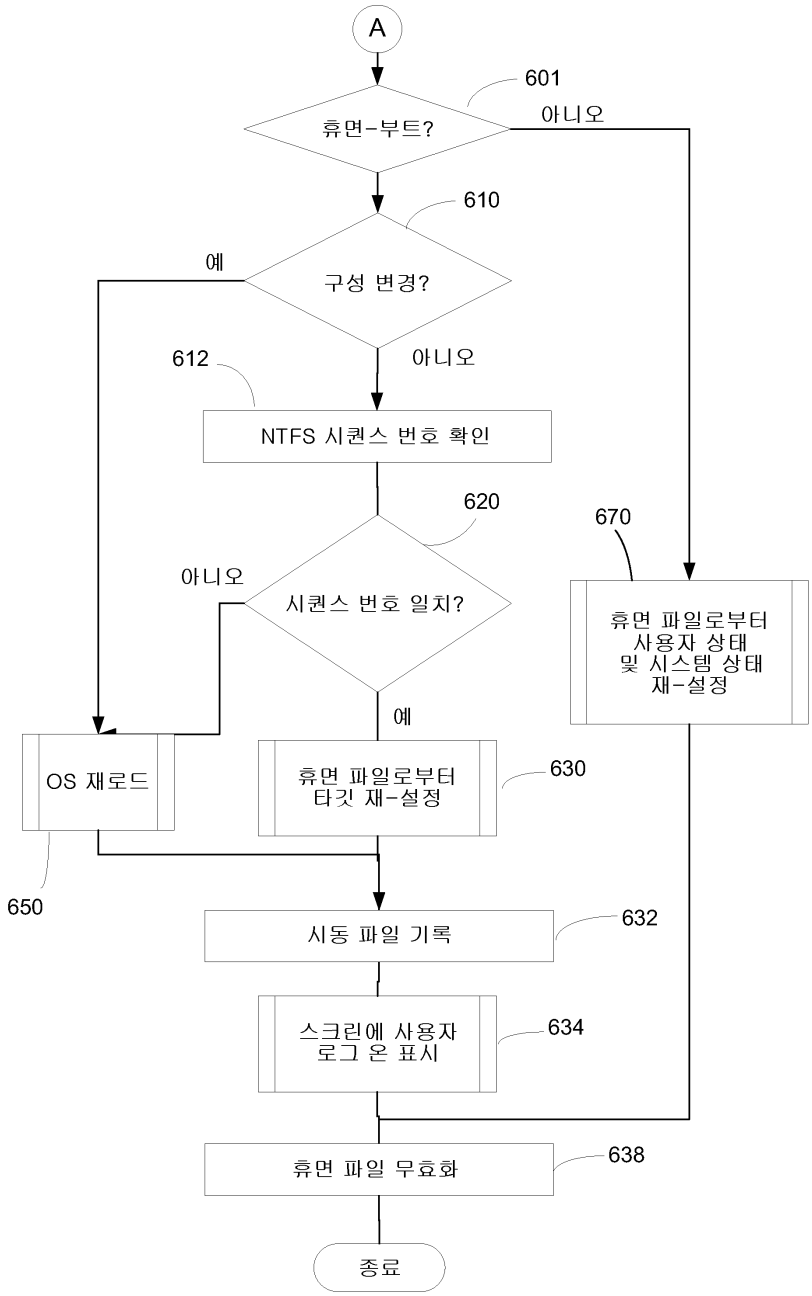
도면4


시작
부트 메모리로부터 인출 및 실행
410
412
휴면 파일 존재?
예
A
아니오
450
OS 로더 실행
420
OS 이미지 구성
422
장치 검출
424
드라이버 로더 실행
426
드라이버 시작
428
서비스 시작
430
애플리케이션 실행
432
440
리부트 필요?
예
442
애플리케이션
등록을 위해 API 호출
아니오
시동 시점 기록
444
동작
446

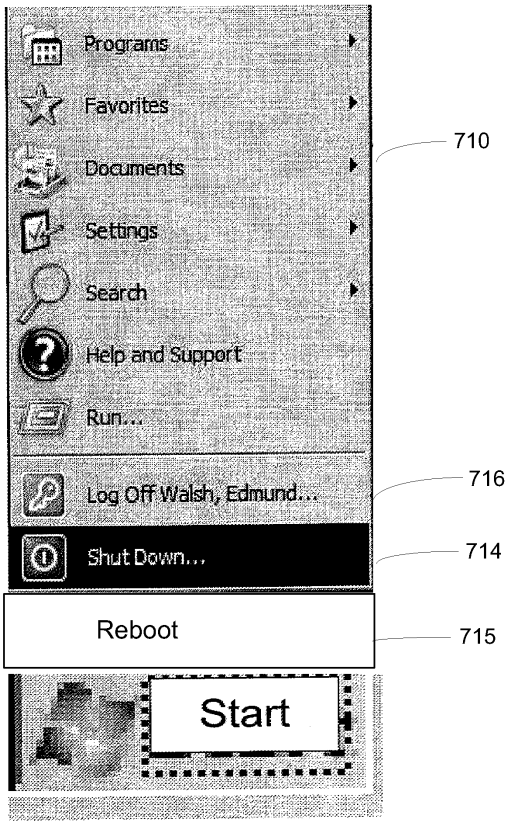
도면5


시작
명령
512
동작
510
사용자 세션(들) 종료
514
516
리부트 요청?
예
아니오
518
재개 시점이
바람직한가?
아니오
예
520
구성 변경?
예
522
더티 데이터 플러쉬
NTFS 시퀀스 번호 기록
524
휴면 파일 저장
526
530
셧다운
종료

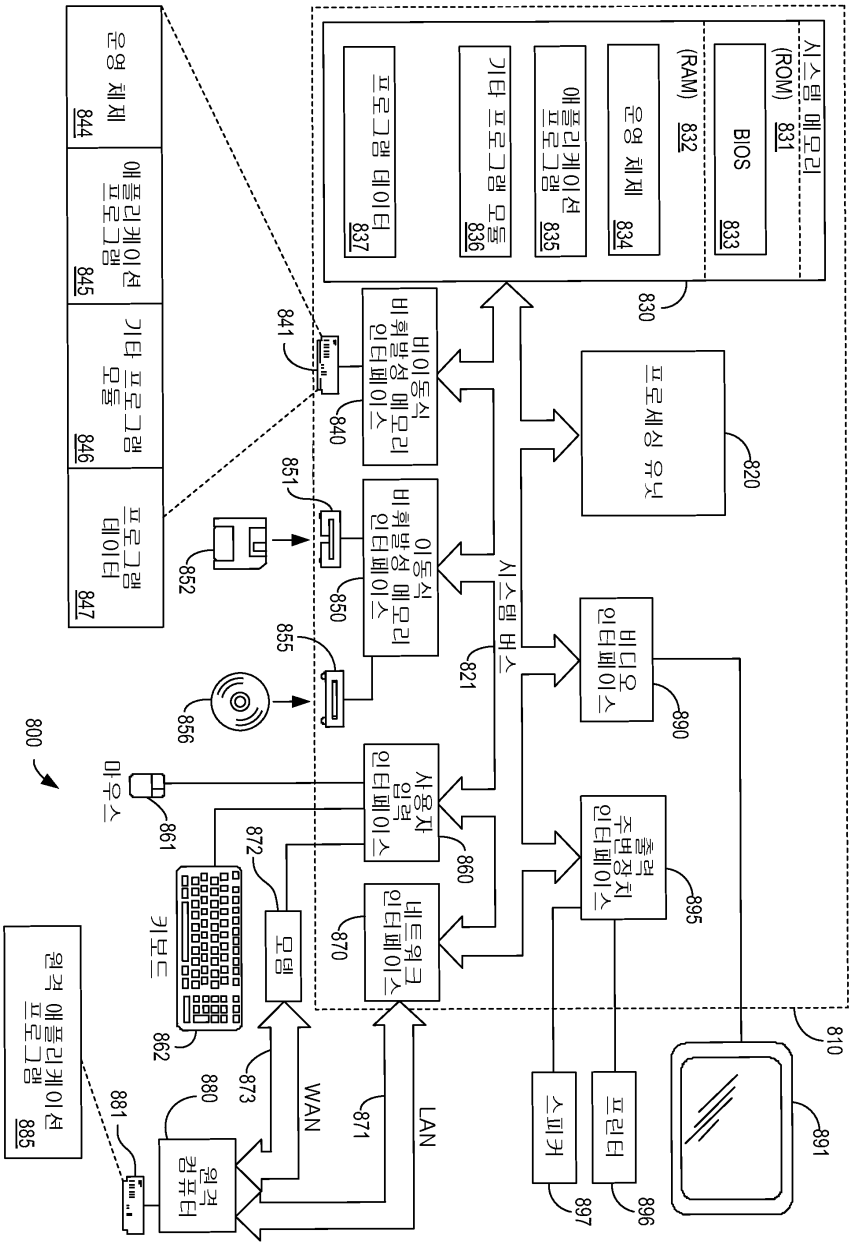
도면6


A
601
휴면-부트?
아니오
610
예
구성 변경?
612
아니오
NTFS 시퀀스 번호 확인
620
670
아니오
시퀀스 번호 일치?
휴면 파일로부터
사용자 상태
및 시스템 상태
재-설정
예
OS 재로드
휴면 파일로부터
타깃 재-설정
630
650
시동 파일 기록
632
스크린에 사용자
로그 온 표시
634
휴면 파일 무효화
638
종료

도면7


Programs
Favorites
Documents
Settings
Search
Help and Support
Run...
Log Off Walsh, Edmund...
Shut Down...
Reboot
Start
710
716
714
715

도면8


시스템 메모리
(ROM) 831
BIOS 833
(RAM) 832
운영 체제 834
애플리케이션 프로그램 835
기타 프로그램 모듈 836
프로그램 데이터 837
830
프로세싱 유닛
820
시스템 버스
821
비휘발성 메모리 인터페이스
840
841
이동식 비휘발성 메모리 인터페이스
850
851
852
855
856
비디오 인터페이스
890
출력 주변장치 인터페이스
895
네트워크 인터페이스
870
사용자 입력 인터페이스
860
마우스
861
872
키보드
862
873
모뎀
880
881
원격 컴퓨터
WAN
871
LAN
원격 애플리케이션 프로그램
885
스피커
897
프린터
896
891
810
800
운영 체제
844
애플리케이션 프로그램
845
기타 프로그램 모듈
846
프로그램 데이터
847