



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0035564
(43) 공개일자 2008년04월23일

(51) Int. Cl.

G06F 9/06 (2006.01) G06F 3/00 (2006.01)
G06F 3/00 (2006.01)

(21) 출원번호 10-2008-7000500

(22) 출원일자 2008년01월08일

심사청구일자 없음

번역문제출일자 2008년01월08일

(86) 국제출원번호 PCT/US2006/026860

국제출원일자 2006년07월10일

(87) 국제공개번호 WO 2007/008880

국제공개일자 2007년01월18일

(30) 우선권주장

11/177,079 2005년07월08일 미국(US)

(71) 출원인

마이크로소프트 코포레이션

미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
마이크로소프트 웨이

(72) 발명자

벤-즈비, 니르

미국 98052-6399 워싱턴주 레드몬드 원 마이크로
소프트 웨이

(74) 대리인

양영준, 백만기

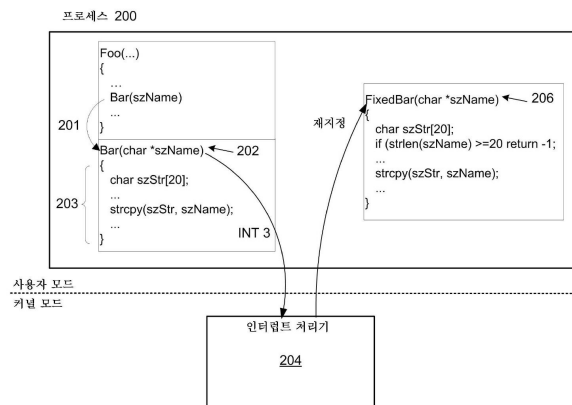
전체 청구항 수 : 총 20 항

(54) 실행 프로세스에서 코드 실행 경로를 재지정하는 방법 및 컴퓨터 판독가능 매체

(57) 요약

실행 프로세스에서 코드 실행 경로를 재지정(redirect)하기 위한 메카니즘이 제공된다. 코드 경로에 1 바이트 인터럽트 명령어(예를 들면, INT 3)가 삽입된다. 인터럽트 명령어는 컨트롤(control)을 커널 처리기에 전달하고, 대체 함수가 실행된 이후에, 복귀하여 이 프로세스를 실행하는 것을 계속한다. 대체 함수는 커널 처리기에 액세스될 수 있는 메모리 공간에 상주한다. 재지정 메카니즘은 실행 프로세스를 실행하고 있는 컴퓨팅 장치를 재부팅할 필요없이 적용될 수 있다. 또한, 재지정 메카니즘은 본래의 코드의 1 바이트를 초과하여 겹쳐 쓰기되지 않으면서 적용될 수 있다.

대표도



특허청구의 범위

청구항 1

실행 프로세스에서 코드 실행 경로를 재지정(redirect)하는 방법으로서,
 상기 코드 실행 경로에 명령어를 주입하는 단계,
 컨트롤(control)을 커널 처리기에 전달하는 단계,
 상기 커널 처리기에 의해 호출된 대체 함수를 실행하는 단계, 및
 상기 코드 실행 경로로 복귀하는 단계
 를 포함하는 실행 프로세스에서 코드 실행 경로를 재지정하는 방법.

청구항 2

제1항에 있어서,
 상기 명령어는 인터럽트(interrupt)인 방법.

청구항 3

제2항에 있어서,
 상기 인터럽트는 INT 3 인터럽트 명령어이며,
 상기 인터럽트 명령어는 길이가 1 바이트인 방법.

청구항 4

제2항에 있어서,
 상기 커널 처리기는 상기 인터럽트 명령어로부터의 복귀가 상기 대체 함수로 계속되게 하는 메카니즘을 포함하는 방법.

청구항 5

제1항에 있어서,
 상기 코드 경로의 본래의 코드의 1 바이트를 초과하여 겹쳐쓰기되지(overwritten) 않도록 상기 명령어를 삽입하는 단계를 더 포함하는 방법.

청구항 6

제1항에 있어서,
 상기 커널 함수에 의해 액세스될 수 있는 메모리 공간으로 상기 대체 함수를 로딩하는 단계를 더 포함하는 방법.

청구항 7

제1항에 있어서,
 상기 실행 프로세스를 실행하고 있는 컴퓨팅 장치를 재부팅할 필요없이 상기 방법을 수행하는 단계를 더 포함하는 방법.

청구항 8

인터럽트를 이용하여 기존의 함수를 대체함으로써 코드 실행 경로를 변경하는 방법으로서,
 상기 기존의 함수에 상기 인터럽트를 주입하는 단계,

컨트롤을 커널 처리기에 전달하는 단계,
 상기 커널 처리기에 의해 호출된 대체 함수를 실행하는 단계, 및
 상기 코드 실행 경로로 복귀하는 단계
 를 포함하는 인터럽트를 이용하여 기존의 함수를 대체함으로써 코드 실행 경로를 변경하는 방법.

청구항 9

제8항에 있어서,
 상기 인터럽트는 INT 3 인터럽트 명령어이며,
 상기 인터럽트 명령어는 길이가 1 바이트인 방법.

청구항 10

제9항에 있어서,
 상기 커널 처리기는 상기 인터럽트 명령어로부터의 복귀가 상기 대체 함수로 계속되게 하는 메커니즘을 포함하는 방법.

청구항 11

제8항에 있어서,
 상기 코드 경로의 본래의 코드의 1 바이트를 초과하여 겹쳐쓰기되지 않도록 상기 인터럽트를 삽입하는 단계를 더 포함하는 방법.

청구항 12

제8항에 있어서,
 상기 커널 함수에 의해 액세스될 수 있는 메모리 공간으로 상기 대체 함수를 로딩하는 단계를 더 포함하는 방법.

청구항 13

제8항에 있어서,
 상기 기존의 함수를 실행하는 프로세스가 실행되고 있는 컴퓨팅 장치를 재부팅할 필요없이 상기 방법을 수행하는 단계를 더 포함하는 방법.

청구항 14

실행 프로세스에서 코드 실행 경로를 재지정하기 위한 컴퓨터 실행가능 명령어를 포함하는 컴퓨터 판독가능 매체로서,
 상기 컴퓨터 실행가능 명령어는
 상기 코드 실행 경로에 명령어를 주입하는 단계,
 컨트롤을 커널 처리기에 전달하는 단계,
 상기 커널 처리기에 의해 호출된 대체 함수를 실행하는 단계, 및
 상기 코드 실행 경로로 복귀하는 단계
 를 포함하는 방법을 수행하는, 실행 프로세스에서 코드 실행 경로를 재지정하기 위한 컴퓨터 실행가능 명령어를 포함하는 컴퓨터 판독가능 매체.

청구항 15

제14항에 있어서,

상기 명령어는 인터럽트인 컴퓨터 판독가능 매체.

청구항 16

제15항에 있어서,

상기 인터럽트는 INT 3 인터럽트 명령어이며,

상기 인터럽트 명령어는 길이가 1 바이트인 컴퓨터 판독가능 매체.

청구항 17

제15항에 있어서,

상기 커널 처리기는 상기 인터럽트 명령어로부터의 복귀가 상기 대체 함수로 계속되게 하는 메카니즘을 포함하는 컴퓨터 판독가능 매체.

청구항 18

제14항에 있어서,

상기 코드 경로의 본래의 코드의 1 바이트를 초과하여 겹쳐쓰기되지 않도록 상기 명령어를 삽입하기 위한 명령어를 더 포함하는 컴퓨터 판독가능 매체.

청구항 19

제14항에 있어서,

상기 커널 함수에 의해 액세스될 수 있는 메모리 공간으로 상기 대체 함수를 로딩하기 위한 명령어를 더 포함하는 컴퓨터 판독가능 매체.

청구항 20

제14항에 있어서,

상기 실행 프로세스를 실행하고 있는 컴퓨팅 장치를 재부팅할 필요 없이 상기 방법을 수행하기 위한 명령어를 더 포함하는 컴퓨터 판독가능 매체.

명세서

기술분야

<1> 본 발명은 일반적으로 컴퓨터 소프트웨어 분야에 관한 것으로, 보다 상세히는 메모리에서 실행되는 프로세스를 업데이트하는 방법에 관한 것이다.

배경기술

<2> 종종 실행 모듈의 초기 온-디스크 이미지를 변경하지 않거나 컴퓨터를 다시 시작할 필요 없이 실행 프로세스에서 코드 실행 경로를 변경하는 것이 요구된다. 이를 수행하는 방식 중 하나는 "핫패칭(Hotpatching)" 메카니즘을 통하는 것이다. 핫패칭은 소프트웨어 업데이트로부터의 코드를 실행 프로세스에 자동으로 삽입함으로써 사용자들이 자신의 컴퓨터를 다시 시작할 필요 없이 소프트웨어 업데이트를 설치할 수 있게 하는 인-메모리 패칭 메카니즘이다. 이는, 시스템 파일이 사용중인 동안에도 이들이 업데이트될 수 있음을 의미한다.

<3> 예를 들면, 취약한 함수의 시작부에 JMP 명령어를 주입함으로써 실행되는 프로세스 내의 취약한 함수를 건너뛸 수 있다. 이 취약한 함수가 호출될 때, 이 함수는 핫패칭 메카니즘에 의해 프로세스 공간에 또한 로딩되는 새로운 함수로 점프한다. 이러한 접근법에 있어서의 문제점은 주입된 JMP 명령어가 예상치 못한 행위에 이르게 되도록 복수의 명령어 위에 겹쳐질 수 있다는 점이다. 핫패칭의 경우, 취약한 함수의 시작부가 처음 5바이트(1바이트 opcode, 2바이트 opcode, 2바이트 opcode)의 3개의 어셈블리 opcode를 포함한다면, JMP 주입은 모두 5바이트를 대체할 것이다. 프로세서가 첫 번째 바이트 opcode를 실행시키고 있어, 주입이 다음 2개의 opcode를 변경했다면, 예상치 못한 프로세스 행위를 발생시킬 수 있다.

발명의 상세한 설명

- <4> 실행 프로세스에서 코드 실행 경로를 재지정(redirect)하기 위한 메카니즘이 제공된다. 코드 경로에 1 바이트 인터럽트 명령어(예를 들면, INT 3)가 삽입된다. 인터럽트 명령어는 컨트롤(control)을 커널 처리기에 전달하고, 커널 처리기는, 대체 함수를 실행한 후에, 복귀하여 이 프로세스를 실행하는 것을 계속한다. 대체 함수는 커널 처리기에 액세스될 수 있는 메모리 공간에 상주한다. 재지정 메카니즘은 실행 프로세스를 실행하고 있는 컴퓨팅 장치를 재부팅할 필요없이 적용될 수 있다. 또한, 재지정 메카니즘은 본래의 코드의 1 바이트를 초과하여 겹쳐쓰기되지 않으면서 적용될 수 있다.
- <5> 본 발명의 추가적인 특징 및 이점은 첨부된 도면을 참조하여 이루어진 예시적인 실시예의 이하의 상세한 설명으로부터 명백해질 것이다.

실시예

- <8> 앞선 발명의 상세한 설명 섹션 및 이하의 바람직한 실시예의 상세한 설명은 첨부된 도면과 관련하여 검토될 때 보다 잘 이해된다. 본 발명을 예시하기 위하여, 이들 도면에서는 본 발명의 예시적인 구조가 도시되지만; 본 발명은 개시된 특정 방법 및 수단에 한정되지 않는다.
- <9> **예시적인 컴퓨팅 환경**
- <10> 도 1은 본 발명이 구현되기에 적합한 컴퓨팅 시스템 환경(100)의 일례를 도시하고 있다. 컴퓨팅 시스템 환경(100)은 적합한 컴퓨팅 환경의 일례에 불과하며, 본 발명의 용도 또는 기능성의 범위에 관해 어떤 제한을 암시하고자 하는 것이 아니다. 컴퓨팅 환경(100)이 예시적인 운영 환경(100)에 도시된 컴포넌트들 중 임의의 하나 또는 그 컴포넌트들의 임의의 조합과 관련하여 어떤 의존성 또는 요구사항을 갖는 것으로 해석되어서는 안된다.
- <11> 본 발명은 많은 기타 범용 또는 특수 목적의 컴퓨팅 시스템 환경 또는 구성에서 동작할 수 있다. 본 발명에서 사용하는 데 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경 및/또는 구성의 예로는 퍼스널 컴퓨터, 서버 컴퓨터, 핸드-헬드 또는 랩톱 장치, 멀티프로세서 시스템, 마이크로프로세서 기반 시스템, 셋톱 박스, 프로그램가능한 가전제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 상기 시스템들이나 장치들 중 임의의 것을 포함하는 분산 컴퓨팅 환경, 기타 등등이 있지만 이에 제한되는 것은 아니다.
- <12> 본 발명은 일반적으로 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터 실행가능 명령어와 관련하여 기술될 것이다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 개체, 컴포넌트, 데이터 구조 등을 포함한다. 본 발명은 또한 통신 네트워크 또는 기타 데이터 전송 매체를 통해 연결되어 있는 원격 처리 장치들에 의해 태스크가 수행되는 분산 컴퓨팅 환경에서 실시될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈 또는 기타 데이터는 메모리 저장 장치를 비롯한 로컬 및 원격 컴퓨터 저장 매체 둘 다에 위치할 수 있다.
- <13> 도 1과 관련하여, 본 발명을 구현하는 예시적인 시스템은 컴퓨터(110) 형태의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(110)의 컴포넌트들은 처리 장치(120), 시스템 메모리(130), 및 시스템 메모리를 비롯한 각종 시스템 컴포넌트들을 처리 장치(120)에 연결시키는 시스템 버스(121)를 포함할 수 있지만 이에 제한되는 것은 아니다. 시스템 버스(121)는 메모리 버스 또는 메모리 컨트롤러, 주변 장치 버스 및 각종 버스 아키텍처 중 임의의 것을 이용하는 로컬 버스를 비롯한 몇몇 유형의 버스 구조 중 어느 것이라도 될 수 있다. 예로서, 이러한 아키텍처는 ISA(Industry Standard Architecture) 버스, MCA(Micro Channel Architecture) 버스, EISA(Enhanced ISA) 버스, VESA(Video Electronics Standard Association) 로컬 버스, (메자닌 버스(Mezzanine bus)로도 알려진) PCI(Peripheral Component Interconnect) 버스, PCI-익스프레스, 및 SM버스(Systems Management Bus)를 포함하지만 이에 제한되는 것은 아니다.
- <14> 컴퓨터(110)는 통상적으로 각종 컴퓨터 판독가능 매체를 포함한다. 컴퓨터(110)에 의해 액세스 가능한 매체는 그 어떤 것이든지 컴퓨터 판독가능 매체가 될 수 있고, 이러한 컴퓨터 판독가능 매체는 휘발성 및 비휘발성 매체, 이동식 및 비이동식 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 정보를 저장하는 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 이동식 및 비이동식 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 컴퓨터(110)에 의해 액세스되고 원하는 정보를 저장할 수 있

는 임의의 기타 매체를 포함하지만 이에 제한되는 것은 아니다. 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 모든 정보 전달 매체를 포함한다. "피변조 데이터 신호"라는 용어는, 신호 내에 정보를 인코딩하도록 그 신호의 특성들 중 하나 이상을 설정 또는 변경시킨 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 배선 접속(direct-wired connection)과 같은 유선 매체, 그리고 음향, RF, 적외선, 기타 무선 매체와 같은 무선 매체를 포함한다. 상술된 매체들의 모든 조합이 또한 컴퓨터 판독가능 매체의 영역 안에 포함되는 것으로 한다.

<15> 시스템 메모리(130)는 ROM(131) 및 RAM(132)와 같은 휘발성 및/또는 비휘발성 메모리 형태의 컴퓨터 저장 매체를 포함한다. 시동 중과 같은 때에, 컴퓨터(110) 내의 구성요소들 사이의 정보 전송을 돕는 기본 루틴을 포함하는 기본 입/출력 시스템(BIOS)(133)은 통상적으로 ROM(131)에 저장되어 있다. RAM(132)은 통상적으로 처리 장치(120)가 즉시 액세스 할 수 있고 및/또는 현재 동작시키고 있는 데이터 및/또는 프로그램 모듈을 포함한다. 예로서, 도 1은 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136) 및 프로그램 데이터(137)를 도시하고 있지만 이에 제한되는 것은 아니다.

<16> 컴퓨터(110)는 또한 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 저장매체를 포함할 수 있다. 단지 예로서, 도 1은 비이동식·비휘발성 자기 매체에 기록을 하거나 그로부터 판독을 하는 하드 디스크 드라이브(141), 이동식·비휘발성 자기 디스크(152)에 기록을 하거나 그로부터 판독을 하는 자기 디스크 드라이브(151), CD-ROM 또는 기타 광 매체 등의 이동식·비휘발성 광 디스크(156)에 기록을 하거나 그로부터 판독을 하는 광 디스크 드라이브(155)를 도시한다. 예시적인 운영 환경에서 사용될 수 있는 기타 이동식/비이동식, 휘발성/비휘발성 컴퓨터 저장 매체로는 자기 테이프 카세트, 플래시 메모리 카드, DVD, 디지털 비디오 테이프, 고상(solid state) RAM, 고상 ROM 등이 있지만 이에 제한되는 것은 아니다. 하드 디스크 드라이브(141)는 통상적으로 인터페이스(140)와 같은 비이동식 메모리 인터페이스를 통해 시스템 버스(121)에 접속되고, 자기 디스크 드라이브(151) 및 광 디스크 드라이브(155)는 통상적으로 인터페이스(150)와 같은 이동식 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

<17> 위에서 설명되고 도 1에 도시된 드라이브들 및 이들과 관련된 컴퓨터 저장 매체는, 컴퓨터(110)를 위해, 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 및 기타 데이터를 저장한다. 도 1에서, 예를 들어, 하드 디스크 드라이브(141)는 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146), 및 프로그램 데이터(147)를 저장하는 것으로 도시되어 있다. 여기서 주의할 점은 이들 컴포넌트가 운영 체제(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)와 동일하거나 그와 다를 수 있다는 것이다. 이에 관해, 운영 체제(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146) 및 프로그램 데이터(147)에 다른 번호가 부여되어 있다는 것은 적어도 이들이 다른 사본(copy)이라는 것을 나타내기 위한 것이다. 사용자는 키보드(162) 및 일반적으로 마우스, 트랙볼(trackball) 또는 터치 패드라 칭하는 포인팅 장치(161) 등의 입력 장치를 통해 명령 및 정보를 컴퓨터(110)에 입력할 수 있다. 다른 입력 장치(도시 생략)로는 마이크, 조이스틱, 게임 패드, 위성 안테나, 스캐너 등을 포함할 수 있다. 이들 및 기타 입력 장치는 종종 시스템 버스에 결합된 사용자 입력 인터페이스(160)를 통해 처리 장치(120)에 접속되지만, 병렬 포트, 게임 포트 또는 USB(universal serial bus) 등의 다른 인터페이스 및 버스 구조에 의해 접속될 수도 있다. 모니터(191) 또는 다른 유형의 디스플레이 장치도 비디오 인터페이스(190) 등의 인터페이스를 통해 시스템 버스(121)에 접속될 수 있다. 모니터 외에, 컴퓨터는 스피커(197) 및 프린터(196) 등의 기타 주변 출력 장치를 포함할 수 있고, 이들은 출력 주변장치 인터페이스(195)를 통해 접속될 수 있다.

<18> 컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 사용하여 네트워크화된 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 또 하나의 퍼스널 컴퓨터, 서버, 라우터, 네트워크 PC, 피어 장치 또는 기타 통상의 네트워크 노드일 수 있고, 통상적으로 컴퓨터(110)와 관련하여 상술된 구성요소들의 대부분 또는 그 전부를 포함하지만, 도 1에는 메모리 저장 장치(181)만이 도시되어 있다. 도시된 논리적 접속으로는 LAN(171) 및 WAN(173)이 있지만, 기타 네트워크를 포함할 수도 있다. 이러한 네트워킹 환경은 사무실, 전사적 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

<19> LAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워킹 환경에서 사용될 때, 컴퓨터(110)는 통상적으로 인터넷과 같은 WAN(173)을 통해 통신을 설정하기 위한 모뎀(172) 또는 기타 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(172)은 사용자 입력 인터페이스(160) 또는 기타 적절한 메커니즘을 통해 시스템 버스(121)에 접속될 수 있다. 네트워크화된 환경에서, 컴퓨터(110) 또는 그의 일부와 관련하여 기술된 프로그램 모듈은 원격 메모리 저장 장치에 저장될 수 있다.

예로서, 도 1은 원격 애플리케이션 프로그램(185)이 원격 장치(181)에 있는 것으로 도시하고 있지만 이에 제한되는 것은 아니다. 도시된 네트워크 접속은 예시적인 것이며 이 컴퓨터들 사이에 통신 링크를 설정하는 기타 수단이 사용될 수 있다는 것을 이해할 것이다.

<20> 예시적인 실시예

<21> 본 발명은 바람직하게는 코드 경로에 삽입되는 1바이트 인터럽트 명령어(예를 들면, INT3)를 이용함으로써 예상치 못한 행위에 이르게 되지 않도록 메모리 내에서 실행되는 프로세스의 코드 실행 경로를 재지정(redirect)하기 위한 메카니즘에 관한 것이다. 인터럽트 명령어는, 대체 함수를 실행한 다음에 복귀하여 프로세스를 실행하는 것을 계속하는 커널 처리기에 컨트롤을 전달한다.

<22> 도 2를 참조하자면, 메모리에서 실행되는 예시적인 프로세스(200)가 도시된다. 본 발명에 따르면, 메모리에서 실행되는 프로세스(200)의 실행 경로(201)(예를 들면, 대체될 취약한 함수의 시작부)는 기존의 명령어(202)를 1바이트 인터럽트 명령어(예를 들면, INT 3)로 겹쳐쓰기함으로써 변경될 수 있는데, 여기에서 본래의 코드(203)의 나머지 부분은 변경되지 않은 상태를 유지한다. INT 3은 통상적으로 다른 코드가 실행되기 위해 실행을 중지시키는 디버거로의 트랩(trap)으로서 이용된다.

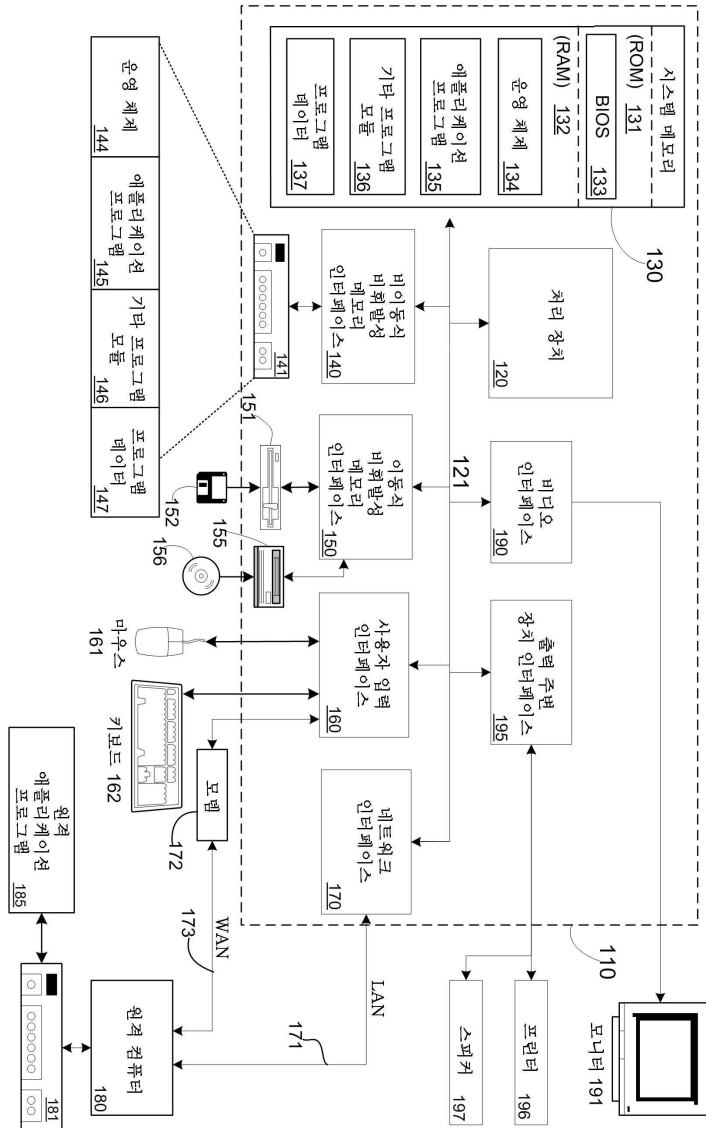
<23> 인터럽트 명령어는 커널 처리기(204)가 호출되도록 할 것이다. 그 인터럽트에 대한 커널 처리기(204)는 인터럽트로부터의 복귀가 본래의 함수로 복귀하는 대신에 새로운 명령어(206)(예를 들면, 대체 함수)로 계속하도록 할 메카니즘을 포함한다. 새로운 명령어(206)는 커널 처리기(204)에 알려진 메모리 공간에 위치한다. 인터럽트 명령어가 1 바이트 명령어이기 때문에, 바람직하게도 본 발명은 본래의 코드(203)의 1 바이트를 초과하여 겹쳐쓰기하지 않는 코드 전환을 위한 메카니즘을 제공한다.

<24> 본 발명은 각종 도면들의 바람직한 실시예에 따라 기술되었지만, 다른 유사한 실시예들이 이용될 수 있거나 기술된 실시예에 본 발명을 벗어나지 않고 본 발명의 동일한 기능을 수행하기 위한 수정 및 추가가 이루어질 수 있다고 이해되어야 한다. 예를 들면, 당업자들은 본 명세서에 기술된 본 발명이 유선이건 무선이건 간에 임의의 컴퓨팅 장치 또는 환경에 적용될 수 있으며 통신 네트워크를 통해 접속되고 이러한 네트워크를 통하여 대화하는 임의의 개수의 컴퓨팅 장치에 적용될 수 있다고 인식할 것이다. 또한, 특히 복수의 무선 네트워크 장치가 급격히 증가하기를 계속하기 때문에, 핸드헬드 장치 운영 체제 및 다른 애플리케이션 특정 운영 체제를 포함하는 각종 컴퓨터 플랫폼을 고려함이 강조되어야 한다. 또한, 본 발명은 복수의 프로세싱 칩 또는 장치에서 또는 이들 간에 구현될 수 있으며, 복수의 장치들 간의 저장 장치가 마찬가지로 영향을 받을 수 있다. 그러므로, 본 발명은 임의의 하나의 실시예에 제한되어야 하는 것이 아니라, 첨부된 특허 청구 범위에 따른 범위 및 범주로 해석되어야 한다.

도면의 간단한 설명

<6> 도 1은 본 발명의 양태가 구현될 수 있는 예시적인 컴퓨팅 환경을 도시하는 블록도.

<7> 도 2는 본 발명에 따라 수행되는 예시적인 프로세스를 도시하는 도면.

컴퓨팅 환경
100

도면2

