(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2012년01월31일
(11) 등록번호 10-1109361
(24) 등록일자 2012년01월17일

(51) Int. Cl.
G06F 1/00 (2006.01) *G06F 9/00* (2006.01)
(21) 출원번호 10-2004-0072652
(22) 출원일자 2004년09월10일
심사청구일자 2009년09월03일
(65) 공개번호 10-2005-0039548
(43) 공개일자 2005년04월29일
(30) 우선권주장
10/693,407 2003년10월24일 미국(US)
(56) 선행기술조사문헌
Karl Heins, Operating System Security: Microsoft Palladium, February 3, 2003
England, etc, A TRUST OPEN PLATFORM, IEEE, 2003.07.
US20020194241 A1
US5822435 A
전체 청구항 수 : 총 18 항

(73) 특허권자
마이크로소프트 코포레이션
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원 마이크로소프트 웨이
(72) 발명자
윌만브란엠.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
츄크리스틴엠.
미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내
(뒷면에 계속)
(74) 대리인
제일특허법인

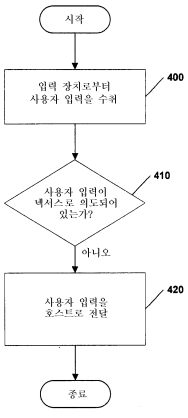
심사관 : 목승균

(54) 고 확실성 실행 환경을 갖는 시스템 내의 신뢰되는에이전트에 안전한 입출력을 제공하는 방법

(57) 요 약

제1의 호스트 오퍼레이팅 시스템이 제2의 고 확실성 오퍼레이팅 시스템(넥서스)과 함께 사용되는 사용자 입출력에 보안을 제공하는 기술이 개시되는데, 제1 시스템은 하부구조의 적어도 일부를 제2 시스템에 제공한다. 신뢰되는 UI 엔진은 신뢰되는 입력 관리자 및 신뢰되는 출력 관리자를 갖는다. 신뢰되는 입력 관리자는 신뢰되는 입력에 대한 액세스를 제어하고, 암호해독된 입력을 적절한 오퍼레이팅 시스템으로, 또는 넥서스에서 실행되는 적절한 프로세스로 분산시킨다. 신뢰되는 출력 관리자는 디스플레이에 대한 출력을 관리하고, 넥서스 내의 신뢰되는 에이전트들로 하여금 출력-장치-의존형 상세를 인식할 필요없이 디스플레이용의 데이터를 출력할 수 있게 한다.

대 표 도 - 도4



(72) 발명자

래이케네스디.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

로버츠폴씨.

미국 98052 워싱턴주 레드몬드 원 마이크로소프트 웨이 마이크로소프트 코포레이션 내

특허청구의 범위

청구항 1

보안 실행 환경(secured execution environment) 및 제2의 실행 환경을 포함하는 시스템 상에서 안전한 사용자 인터페이스를 상기 보안 실행 환경에 제공하기 위한 방법으로서,

상기 보안 실행 환경 또는 상기 제2의 실행 환경에 대하여 의도되는 사용자 입력 장치로부터 암호화된 사용자 입력을 수취하는 단계;

상기 암호화된 사용자 입력을 암호해독하는 단계;

적어도 하나의 그래픽 사용자 요소(graphical user element)들 중에서 포커스를 갖는 그래픽 사용자 요소를 결정하는 단계;

상기 그래픽 사용자 요소를 가지는 프로세스가 상기 보안 실행 환경에 있는지 상기 제2의 실행 환경에 있는지 여부를 결정하는 단계;

상기 그래픽 사용자 요소를 가지는 상기 프로세스가 상기 보안 실행 환경에 있는지 상기 제2의 실행 환경에 있는지 여부에 기초하여, 상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되는지 여부를 결정하는 단계;

상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되지 않는 경우, 상기 암호해독된 사용자 입력을 상기 제2의 실행 환경으로 전송하는 단계;

상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되는 경우, 상기 암호해독된 사용자 입력을 위한 상기 보안 실행 환경 내의 특정 목적지 엔티티를 결정하고, 상기 암호해독된 사용자 입력을 상기 특정 목적지 엔티티로 전송하는 단계;

상기 제2의 실행 환경 내는 아닌, 상기 보안 실행 환경 내의 특정 소스 엔티티로부터의 출력을 수취하는 단계; 및

상기 출력을 출력 장치로 안전하게 전송하는 단계

를 포함하는 방법.

청구항 2

제1항에 있어서,

상기 사용자 입력 장치로부터 암호화된 사용자 입력을 수취하는 단계는 상기 사용자 입력과의 안전한 통신 채널을 확립하는 단계를 포함하는, 방법.

청구항 3

제1항에 있어서,

상기 사용자 입력 장치로부터 암호화된 사용자 입력을 수취하는 단계는 상기 사용자 입력을 검증하는 단계를 포함하는, 방법.

청구항 4

제1항에 있어서,

상기 암호해독된 사용자 입력을 상기 특정 목적지 엔티티로 전송하는 단계는 상기 암호해독된 사용자 입력을 해석하는 단계를 포함하는, 방법.

청구항 5

제1항에 있어서,

상기 출력을 상기 출력 장치로 안전하게 전송하는 단계는 상기 출력의 데이터를 암호화하는 단계를 포함하는,

방법.

청구항 6

제1항에 있어서,

상기 출력을 상기 출력 장치로 안전하게 전송하는 단계는 상기 출력을 커튼드 메모리(curtained memory)로 전송하는 단계를 포함하는, 방법.

청구항 7

제1항에 있어서,

상기 출력은 데이터 부분을 가지고,

상기 출력을 상기 출력 장치로 안전하게 전송하는 단계는 상기 출력의 상기 데이터 부분을 암호화하는 단계를 포함하는, 방법.

청구항 8

보안 실행 환경 및 제2의 실행 환경을 포함하는 시스템 상에서 안전한 사용자 인터페이스를 상기 보안 실행 환경에 제공하기 위한 컴퓨터 실행가능 명령들을 포함하는 컴퓨터 판독가능 저장 매체로서,

상기 컴퓨터 실행가능 명령들은,

상기 보안 실행 환경 또는 상기 제2의 실행 환경에 대하여 의도되는 사용자 입력 장치로부터 암호화된 사용자 입력을 수취하는 단계;

상기 암호화된 사용자 입력을 암호해독하는 단계;

적어도 하나의 그래픽 사용자 요소들 중에서 포커스를 갖는 그래픽 사용자 요소를 결정하는 단계;

상기 그래픽 사용자 요소를 가지는 프로세스가 상기 보안 실행 환경에 있는지 상기 제2의 실행 환경에 있는지 여부를 결정하는 단계;

상기 그래픽 사용자 요소를 가지는 상기 프로세스가 상기 보안 실행 환경에 있는지 상기 제2의 실행 환경에 있는지 여부에 기초하여, 상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되는지 여부를 결정하는 단계;

상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되지 않는 경우, 상기 암호해독된 사용자 입력을 상기 제2의 실행 환경으로 전송하는 단계;

상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되는 경우, 상기 암호해독된 사용자 입력을 위한 상기 보안 실행 환경 내의 특정 목적지 엔티티를 결정하고, 상기 암호해독된 사용자 입력을 상기 특정 목적지 엔티티로 전송하는 단계;

상기 제2의 실행 환경 내는 아닌, 상기 보안 실행 환경 내의 특정 소스 엔티티로부터의 출력을 수취하는 단계; 및

상기 출력을 출력 장치로 안전하게 전송하는 단계

를 포함하는 동작들을 수행하기 위한 것인, 컴퓨터 판독가능 저장 매체.

청구항 9

제8항에 있어서,

상기 사용자 입력 장치로부터 암호화된 사용자 입력을 수취하는 단계는 상기 사용자 입력과의 안전한 통신 채널을 확립하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 10

제8항에 있어서,

상기 사용자 입력 장치로부터 암호화된 사용자 입력을 수취하는 단계는 상기 사용자 입력을 검증하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 11

제8항에 있어서,

상기 암호해독된 사용자 입력을 상기 특정 목적지 엔티티로 전송하는 단계는 상기 암호해독된 사용자 입력을 해석하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 12

제8항에 있어서,

상기 출력은 데이터 부분을 가지고,

상기 출력을 상기 출력 장치로 안전하게 전송하는 단계는 상기 출력의 상기 데이터 부분을 암호화하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 13

제8항에 있어서,

상기 출력을 상기 출력 장치로 안전하게 전송하는 단계는 상기 출력을 커튼드 메모리로 전송하는 단계를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 14

보안 실행 환경 및 제2의 실행 환경을 포함하는 시스템 상에서 안전한 사용자 인터페이스를 상기 보안 실행 환경에 제공하기 위한 신뢰되는 사용자 인터페이스 엔진(trusted user interface engine)을 구현하기 위한 컴퓨터 실행가능 명령들이 저장된 컴퓨터 판독가능 저장 매체로서,

상기 신뢰되는 사용자 인터페이스 엔진은,

사용자 입력 장치로부터 암호화된 사용자 입력을 수취하고 상기 암호화된 사용자 입력을 암호해독하며, 상기 사용자 입력 장치에 동작가능하게 연결되는, 신뢰되는 입력 서비스 제공자(input trusted service provider);

적어도 하나의 그래픽 사용자 요소들 중에서 포커스를 갖는 그래픽 사용자 요소를 결정하고, 상기 그래픽 사용자 요소를 가지는 프로세스가 상기 보안 실행 환경에 있는지 상기 제2의 실행 환경에 있는지 여부를 결정하고, 상기 그래픽 사용자 요소를 가지는 상기 프로세스가 상기 보안 실행 환경에 있는지 상기 제2의 실행 환경에 있는지 여부에 기초하여, 상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되는지 여부를 결정하며, 상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되지 않는 경우, 상기 암호해독된 사용자 입력을 상기 제2의 실행 환경으로 전송하고, 상기 암호해독된 사용자 입력이 상기 보안 실행 환경에 대하여 의도되는 경우, 상기 암호해독된 사용자 입력을 위한 상기 보안 실행 환경 내의 특정 목적지 엔티티를 결정하고 상기 암호해독된 사용자 입력을 상기 특정 목적지 엔티티로 전송하는, 신뢰되는 입력 관리자(trusted input manager); 및

상기 제2의 실행 환경 내는 아닌, 상기 보안 실행 환경 내의 특정 소스 엔티티로부터의 출력을 수취하고, 상기 출력을 출력 장치로 안전하게 전송하는, 신뢰되는 출력 관리자(trusted output manager)

를 포함하는, 컴퓨터 판독가능 저장 매체.

청구항 15

제14항에 있어서,

상기 신뢰되는 입력 서비스 제공자는 상기 사용자 입력과의 안전한 통신 채널을 확립하는, 컴퓨터 판독가능 저장 매체.

청구항 16

제14항에 있어서,

상기 신뢰되는 입력 서비스 제공자는 상기 사용자 입력을 검증하는, 컴퓨터 판독가능 저장 매체.

청구항 17

제14항에 있어서,

상기 출력은 데이터 부분을 가지고,

상기 신뢰되는 출력 관리자는 상기 출력의 상기 데이터 부분을 암호화하는, 컴퓨터 판독가능 저장 매체.

청구항 18

제14항에 있어서,

상기 신뢰되는 출력 관리자는 상기 출력을 커튼드 메모리로 전송하는, 컴퓨터 판독가능 저장 매체.

청구항 19

삭제

청구항 20

삭제

청구항 21

삭제

청구항 22

삭제

청구항 23

삭제

청구항 24

삭제

청구항 25

삭제

청구항 26

삭제

청구항 27

삭제

청구항 28

삭제

청구항 29

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

청구항 35

삭제

청구항 36

삭제

청구항 37

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

명 세 서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

[0013] 본 발명은 일반적으로 컴퓨터 보안 분야에 관한 것이다. 특히, 본 발명은 단일 컴퓨팅 장치 상에서 복수의 실행 환경들(예를 들어, 오퍼레이팅 시스템들)을 사용하는 것에 관한 것으로, 복수의 실행 환경들 중에서 보다 높은 확실성 실행 환경의 신뢰되는 에이전트에 대한 데이터의 입출력 보전성을 가능케 하는 기술들을 제공한다.

[0014] 현재 컴퓨팅에 있어서, 컴퓨터 상에서 수행될 수 있는 많은 작업들은 일정 레벨의 보안을 요구한다. 보안의 레벨을 제공하기 위해, 여러가지 선택들이 존재한다. 하나는 임의의 안전하지 않을 수 있는 요소들로부터 완전히 분리되어 있는 컴퓨터 상에서 모든 안전한 어플리케이션들을 수행하는 것, 또는 VMM(virtual machine monitor)를 사용하여 단일 컴퓨터 시스템 상에서 운영되는 2가지의 실행 환경들(예를 들어, 오퍼레이팅 시스템들) 사이의 완전한 분리를 허용하는 것이다. 그러나, 이것은 실용적이지 않을 수 있다. 비용 또는 편의상, 안전한 실행 환경은 확실하지 않은 보안을 갖는 어플리케이션들과 자원들을 공유할 필요가 있고, 이러한 어플리케이션들 및 이러한 자원들은 공격자에게 공격받기 쉽다. 또한, VMM이 사용되는 곳에서는, VMM이 기계 및 그 장치들 모두의 전체 가상화(virtualization)를 요구하기 때문에(따라서 VMM이 모든 가능한 장치에 대해 그

자신의 장치 드라이버를 제공하도록 요구함), VMM은 거의 제한없는 각종 장치들이 기계에 추가될 수 있는 오픈 아키텍처 기계에 적합하지 않다.

[0015] 2가지 실행 환경들 사이에서 자원들을 공유하는 능력을 제공하는 한가지 방법은, 하나의 기계에 있는 대부분의 프로세스들 및 장치들을 제어하는 하나의 "메인" 오퍼레이팅 시스템이 존재하고, 또한 제2 오퍼레이팅 시스템이 존재하는 컴퓨터 시스템을 제공하는 것이다. 이러한 제2 오퍼레이팅 시스템은 특정의 제한된 작업을 수행하는 메인 오퍼레이팅 시스템과 나란히 소형이고 제한된 목적의 오퍼레이팅 시스템이다. 오퍼레이팅 시스템을 "소형" 또는 "제한된 목적"으로 만드는 한가지 방법은 소형 오퍼레이팅 시스템이 "메인" 오퍼레이팅 시스템으로부터 특정 하부구조(예를 들어, 스케쥴링 설비, 메모리 관리자, 장치 드라이버 등)를 빌리게 하는 것이다. VMM은 하나의 오퍼레이팅 시스템을 다른 것과 효과적으로 분리시키기 때문에, VMM을 이용하는 이러한 인프라구조의 공유는 실용적이지 않다.

[0016] 특정의 다른 기술들은 VMM을 사용하지 않고 동일한 시스템들이 동일한 기계에 나란히 존재하게 하는 것이다. 하나의 이와 같은 기술은 다른 오퍼레이팅 시스템에 대한 "호스트"로서 작용하는 하나의 오퍼레이팅 시스템을 갖는 것이다.("호스트"가 호스팅하는 오퍼레이팅 시스템은 때때로 "게스트"라고 지칭됨) 이러한 경우에, 호스트 오퍼레이팅 시스템은 메모리 및 프로세서 시간과 같은 자원들을 게스트에게 제공한다. 또 다른 이러한 기술은 "엑소커넬(exokernel)"을 사용하는 것이다. 엑소커넬 - VMM과 다름 - 이 전체 기계를 가상화하지는 않지만, 엑소커넬은 특정한 장치들(예를 들어, 프로세서 및 메모리)를 관리하고, 또한 오퍼레이팅 시스템들 사이에서 특정 유형의 상호작용을 관리한다. 엑소커넬이 사용되는 경우라도, 하나의 오퍼레이팅 시스템(예를 들어, "메인" 오퍼레이팅 시스템)이 하부구조의 대부분을 다른 오퍼레이팅 시스템에 제공하는 경우가 될 수 있고, 이 경우에 주 오퍼레이팅 시스템은 "호스트"라 지칭될 수 있고, 보다 작은 오퍼레이팅 시스템은 "게스트"로서 지칭될 수 있다. 호스팅 모델 및 엑소커넬 모델 모두는 인프라 구조의 공유를 지원하는 오퍼레이팅 시스템들 사이에서 유용한 유형의 상호작용을 허용한다.

[0017] 따라서, 이러한 기술들은 적어도 2가지 실행 환경들을 컴퓨터 시스템에 제공하기 위해 사용될 수 있다. 이들 중 하나는 "고 확실성" 오퍼레이팅 시스템일 수 있는데, 본 명세서에서는 "넥서스"라고 지칭된다. 고 신뢰성 오퍼레이팅 시스템은 그 거동에 대해 특정 수준의 확실성을 제공하는 것이다. 예를 들어, 넥서스는 넥서스 외부 세계로 정보를 유출시키지 않도록 보장되는 커튼드(curtained) 메모리를 제공함으로써, 그리고 단지 특정한 증명된 어플리케이션들이 넥서스 하에서 실행되고 커튼드 메모리를 액세스하도록 허용함으로서, 누설되어서는 안되는 비밀 정보(예를 들어 암호화 키들 등)에 의해 동작하도록 채용될 것이다.

[0018] 2가지 실행 환경들을 갖는 컴퓨터 시스템에서, 그 실행 환경들 중 하나는 넥서스인데, 넥서스는 게스트 오퍼레이팅 시스템인 것이 바람직하고, 거동에 대해 확실성 레벨이 동일하지 않은 제2 오퍼레이팅 시스템은 호스트 오퍼레이팅 시스템인 것이 바람직하다. 이것은 넥서스가 가능한 한 소형으로 되는 것을 가능하게 한다. 소형의 넥서스는 넥서스에 의해 제공되는 확실성에서 보다 높은 수준의 확신을 허용한다.

[0019] 그러나, 넥서스의 고 확실성 특징은 넥서스에서 실행되는 프로세스들에 대한 입출력에 대해 고 확실성을 요구하므로, 호스트 오퍼레이팅 시스템으로부터의 프로세스 또는 기타 엔티티는 사용자에 의해 입력되는 데이터, 또는 사용자에게 표시되거나 출력되는 데이터를 판독 또는 변경할 수 없다. 그러나 호스트 오퍼레이팅 시스템이 입출력을 처리하고 그 프로세스를 위해 정보를 넥서스에게 중계하게 하는 것은 넥서스의 고 확실성 특징을 위험하게 할 것이다. 또한, 입력은 입력을 암호화하는 신뢰되는 사용자 입력 장치로부터 나올 것이고, 호스트에게 유출되어서는 안되는 넥서스 내에 보유된 비밀을 사용하여 데이터를 암호해독할 필요가 있을 것이다.

[0020] 사용자에게 표시되고 있는 그래픽 사용자 인터페이스 요소들에서 사용자 이벤트들을 렌더링하고 검출하고 처리하는 등의 입출력(I/O) 기능들은 종종 모든 프로세스들에 대한 공통 자원에 의해 제공된다. 그러나, 이러한 호스트 오퍼레이팅 시스템의 기능을 제공하는 것은 렌더링되는 데이터가 렌더링용 호스트로 패스될 것을 요구한다. 이것은 렌더링을 위해 송신되는 데이터가 데이터에 액세스해서는 안되는 호스트-사이드 엔티티에 의해 판독되거나 변경될 수 있기 때문에 렌더링을 위해 데이터를 배출하는 프로세스의 고 확실성 특징에 가능한 어택 방법을 제공한다. 사용자 이벤트가 발생한 통지에는 동일한 취약성이 존재한다.

[0021] 상기 관점에서 종래 기술의 문제점을 극복하는 시스템에 대한 필요성이 있다.

발명이 이루고자 하는 기술적 과제

[0022] 보안되는 실행 환경 및 제2 실행 환경을 갖는 시스템에 대한 보안되는 실행 환경의 보안은 신뢰되는 UI 엔진을 사용하여 유지된다. 신뢰되는 UI 엔진은 보안되는 실행 환경에 대한 사용자 입력 및 보안되는 실행 환경의 프

로세스로부터 디스플레이 또는 출력 장치로의 출력을 조정한다.

[0023] 일 실시예에서, 신뢰되는 UI 엔진의 일 컴포넌트는 신뢰되는 입력 관리자이다. 신뢰되는 입력 장치로부터 암호화된 입력이 도착하면, 신뢰되는 입력 관리자가 입력을 암호해독한다. 그 후 입력이 보안되는 실행 환경에 존재해야 하는지 또는 제2 실행 환경으로 송신되어야하는지를 결정한다. 암호화된 입력은 보안되는 사용자 환경의 결정 후에만 제2 실행 환경으로 전달된다.

[0024] 신뢰되는 UI 엔진의 일 컴포넌트는 신뢰되는 출력 관리자이다. 출력에 대한 보안을 제공하기 위해, 신뢰되는 출력 관리자는 보안되는 실행 환경의 모든 엔티티들에 대한 접촉 지점으로서 기능한다.

[0025] 발명의 다른 특징들은 후술된다.

발명의 구성 및 작용

[0026] 상기한 요약뿐만 아니라 이하의 바람직한 실시예들의 상세한 설명은 첨부 도면을 참조하는 경우에 보다 잘 이해된다. 본 발명의 설명을 위해, 발명의 예시적인 구성들이 도면에 도시되지만, 본 발명은 개시되어 있는 특정 방법 및 수단에 한정되지 않는다.

[0027] 오퍼레이팅 시스템과 같은 2가지 실행 환경이 단일 기계 상에서 나란히 운영되는 경우, 사용자 입력 및 출력이 오퍼레이팅 시스템에 의해 어떻게 액세스되어야한는지를 결정해야 한다. 또한, 오퍼레이팅 시스템들중 하나는 제2 오퍼레이팅 시스템 또는 제2 오퍼레이팅 시스템에서 실행되는 엔티티에 대한 사용자 입력 또는 그로부터의 사용자 출력을 엑세스하는 것이 보호될 필요가 있을 수 있다. 본 발명은 넥서스 상의 고 확실성 엔티티로 향하거나 그로부터 나오는 사용자 입력 및 출력이 호스트 오퍼레이팅 시스템 엔티티들에 의해 발견되는 것을 보호할 수 있다.

[0028] 예시적인 컴퓨팅 환경

[0029] 도 1은 본 발명의 특징이 구현될 수 있는 예시적인 컴퓨팅 환경의 예를 나타낸다. 컴퓨팅 시스템 환경(100)은 단지 적절한 컴퓨팅 환경의 일 예이며 본 발명의 사용 또는 기능의 범위에 제한을 가하도록 의도된 것은 아니다. 컴퓨팅 환경(100)은 예시적인 오퍼레이팅 환경(100)에 도시된 컴포넌트들 중의 임의의 하나 또는 조합에 관하여 임의의 종속성(dependency) 또는 요구사항(requirement)을 갖는 것으로 해석되어서는 안된다.

[0030] 본 발명은 많은 다른 범용 또는 특수목적 컴퓨팅 시스템 환경들 또는 구성들과 함께 동작될 수 있다. 본 발명과 함께 사용하기에 적합할 수 있는 잘 알려진 컴퓨팅 시스템, 환경, 및/또는 구성의 예로는, 퍼스널 컴퓨터, 서버 컴퓨터, 핸드헬드(hand-held) 또는 랩탑 장치, 멀티프로세서 시스템, 마이크로프로세서-기반 시스템, 셋탑 박스(set top box), 프로그램가능한 가전제품(programmable consumer electronics), 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터, 내장 시스템, 상기의 시스템 또는 장치 중의 임의의 것을 포함하는 분산형 컴퓨팅 환경 등이 포함될 수 있지만, 이에 한정되지 않는다.

[0031] 본 발명은 컴퓨터에 의해 실행되는, 프로그램 모듈과 같은 컴퓨터 실행가능 명령과 일반적으로 관련하여 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 유형을 구현하는 루틴, 프로그램, 오브젝트, 컴포넌트, 데이터 구조 등을 포함한다. 본 발명은 또한 통신 네트워크 또는 다른 데이터 전송 매체를 통해 링크된 원격 프로세싱 장치에 의해 태스크를 수행하는 분산형 컴퓨팅 환경에서 실행될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈 및 그외 데이터는 메모리 저장 장치를 포함하는 국부 및 원격 컴퓨터 저장 매체 모두에 위치할 수 있다.

[0032] 도 1을 참조하면, 본 발명을 구현하기 위한 예시적인 시스템은 컴퓨터(110)의 형태의 범용 컴퓨팅 장치를 포함한다. 컴퓨터(110)의 컴포넌트들로는, 프로세싱 유닛(120), 시스템 메모리(130), 및 시스템 메모리를 포함하는 다양한 시스템 컴포넌트를 프로세싱 유닛(120)에 연결시키는 시스템 버스(121)가 포함될 수 있지만, 이에 한정되는 것은 아니다. 프로세싱 유닛(120)은 멀트 쓰레드된 프로세서에서 지원되는 것과 같은 다수의 논리 프로세싱 유닛을 나타낸다. 시스템 버스(121)는 다양한 버스 아키텍처 중의 임의의 것을 사용하는 로컬 버스, 주변 버스, 및 메모리 버스 또는 메모리 컨트롤러를 포함하는 몇가지 유형의 버스 구조 중의 임의의 것일 수 있다. 예로서, 이러한 아키텍처는 산업 표준 아키텍처(ISA) 버스, 마이크로 채널 아키텍처(MCA) 버스, 인핸스드 ISA(Enhanced ISA; EISA) 버스, 비디오 일렉트로닉스 표준 어소시에이션(VESA) 로컬 버스, 및 (메자닌(Mezzanine) 버스로도 알려진) 주변 컴포넌트 상호접속(PCI) 버스를 포함하지만, 이에 한정되는 것은 아니다. 시스템 버스(121)는 통신 장치 사이에서 점대점 접속, 스위칭 구조 등으로서 구현될 수도 있다.

[0033] 컴퓨터(110)는 통상적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 컴퓨터 판독가능 매체는 컴퓨터(110)에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수 있으며, 휘발성 및 불휘발성 매체, 분리형(removable) 및 비분리형(non-removable) 매체를 둘다 포함한다. 예로서, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있지만, 이에 한정되는 것은 아니다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 휘발성 및 불휘발성, 분리형 및 비분리형 매체를 둘다 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래쉬 메모리 또는 기타 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 기타 광학 디스크 저장장치, 자기 카세트, 자기 테이프, 자기 디스크 저장장치 또는 기타 자기 저장장치, 또는 컴퓨터(110)에 의해 액세스될 수 있고 원하는 정보를 저장하는 데 사용될 수 있는 임의의 기타 매체를 포함할 수 있지만, 이에 한정되지 않는다. 통신 매체는 통상적으로 반송파 또는 기타 전송 메카니즘 등의 변조된 데이터 신호에 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈, 또는 다른 데이터를 구현하며, 임의의 정보 전달 매체를 포함한다. "변조된 데이터 신호"라는 용어는 신호 내에 정보를 인코딩하도록 설정되거나 변환된 특성을 하나 또는 그 이상을 갖는 신호를 의미한다. 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 접속 등의 유선 매체와, 음향, RF, 적외선 및 기타 무선 매체 등의 무선 매체를 포함하지만, 이에 한정되지 않는다. 상술한 것들 중의 임의의 조합이 컴퓨터 판독가능 매체의 범위 내에 포함되어야 한다.

[0034] 시스템 메모리(130)는 ROM(131) 및 RAM(132) 등의 휘발성 및/또는 불휘발성 메모리의 형태의 컴퓨터 저장 매체를 포함한다. 시동중과 같은 때에 컴퓨터(110) 내의 구성요소들간에 정보를 전송하는 것을 돕는 기본 루틴을 포함하는 기본 입출력 시스템(133; BIOS)은 일반적으로 ROM(131)에 저장된다. RAM(132)은 일반적으로 프로세싱 유닛(120)에 즉시 액세스될 수 있고 및/또는 프로세싱 유닛(120)에 의해 현재 작동되는 프로그램 모듈 및/또는 데이터를 포함한다. 예로서, (한정하고자 하는 것은 아님) 도 1은 오퍼레이팅 시스템(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)를 도시한다.

[0035] 컴퓨터(110)는 또한 다른 분리형/비분리형, 휘발성/불휘발성 컴퓨터 저장 매체를 포함할 수 있다. 단지 예로서, 도 1에는 비분리형 불휘발성 자기 매체로부터 판독하거나 그 자기 매체에 기록하는 하드 디스크 드라이브(140), 분리형 불휘발성 자기 디스크(152)로부터 판독하거나 그 자기 디스크에 기록하는 자기 디스크 드라이브(151), 및 CD-ROM 또는 기타 광학 매체 등의 분리형 불휘발성 광학 디스크(156)로부터 판독하거나 그 광학 디스크에 기록하는 광학 디스크 드라이브(155)가 도시되어 있다. 예시적인 오퍼레이팅 환경에서 사용될 수 있는 기타 분리형/비분리형, 휘발성/불휘발성 컴퓨터 저장 매체는 자기 테이프 카세트, 플래쉬 메모리 카드, DVD(Digital versatile disk), 디지털 비디오 테이프, 고체 RAM, 고체 ROM 등을 포함하지만 이에 한정되지 않는다. 하드 디스크 드라이브(141)는 일반적으로 인터페이스(140)와 같은 비분리형 메모리 인터페이스를 통해 시스템 버스(121)에 접속되고, 자기 디스크 드라이브(151) 및 광학 디스크 드라이브(155)는 일반적으로 인터페이스(150)와 같은 분리형 메모리 인터페이스에 의해 시스템 버스(121)에 접속된다.

[0036] 앞서 기술되고 도 1에 도시된 드라이브 및 그 관련 컴퓨터 저장 매체는 컴퓨터(110)를 위한 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 및 기타 데이터의 저장을 제공한다. 도 1에서, 예를 들어, 하드 디스크 드라이브(141)는 오퍼레이팅 시스템(144), 애플리케이션 프로그램(145), 기타 프로그램 모듈(146), 및 프로그램 데이터(147)를 저장하는 것으로 도시된다. 이들 컴포넌트는 오퍼레이팅 시스템(134), 애플리케이션 프로그램(135), 기타 프로그램 모듈(136), 및 프로그램 데이터(137)와 동일할 수도 있고 다를 수도 있다. 오퍼레이팅 시스템(144), 애플리케이션 프로그램(145), 다른 프로그램 모듈(146), 및 프로그램 데이터(147)는 최소한 다른 복사본(different copies)임을 나타내기 위하여 다른 번호를 부여하였다. 사용자는 일반적으로 마우스, 트랙볼, 또는 터치 패드라 불리우는 포인팅 장치(161) 및 키보드(162)와 같은 입력 장치를 통해 컴퓨터(110)에 명령 및 정보를 입력할 수 있다. (도시되지 않은) 기타 입력 장치는 마이크로폰, 조이스틱, 게임 패드, 위성 안테나, 스캐너 등을 포함할 수 있다. 이들 입력 장치 및 그외의 입력 장치는 시스템 버스에 연결된 사용자 입력 인터페이스(160)를 통해 종종 프로세싱 유닛(120)에 접속되지만, 병렬 포트, 게임 포트 또는 유니버설 시리얼 포트(USB) 와 같은 기타 인터페이스 및 버스 구조에 의해 접속될 수 있다. 모니터(191) 또는 다른 유형의 디스플레이 장치는 또한 비디오 인터페이스(190) 등의 인터페이스를 통해 시스템 버스(121)에 접속된다. 모니터외에도, 컴퓨터는 또한 출력 주변 인터페이스(195)를 통해 접속될 수 있는 스피커(197) 및 프린터(196) 등의 기타 주변 출력 장치를 포함할 수 있다.

[0037] 컴퓨터(110)는 원격 컴퓨터(180)와 같은 하나 이상의 원격 컴퓨터로의 논리적 접속을 이용한 네트워크 환경에서 동작할 수 있다. 원격 컴퓨터(180)는 퍼스널 컴퓨터, 서버, 라우터, 네트워크 PC, 피어(peer) 장치, 또는 기타 공통 네트워크 노드일 수 있으며, 비록 도 1 에는 메모리 저장 장치(181)만이 도시되어 있지만, 컴퓨터(110)에

관하여 상술한 구성요소 중 다수 또는 모든 구성요소를 일반적으로 포함할 수 있다. 도 1에 도시된 논리적 접속은 근거리 통신망(LAN; 171) 및 원거리 통신망(WAN; 173)을 포함하지만, 그 외의 네트워크를 포함할 수도 있다. 이러한 네트워크 환경은 사무실, 기업 광역 컴퓨터 네트워크(enterprise-wide computer network), 인트라넷, 및 인터넷에서 일반적인 것이다.

[0038] LAN 네트워크 환경에서 사용되는 경우, 컴퓨터(110)는 네트워크 인터페이스 또는 어댑터(170)를 통해 LAN(171)에 접속된다. WAN 네트워크 환경에서 사용되는 경우, 컴퓨터(110)는 일반적으로 인터넷 등의 WAN(173)을 통해 통신을 구축하기 위한 모뎀(172) 또는 기타 수단을 포함한다. 내장형 또는 외장형일 수 있는 모뎀(172)은 사용자 입력 인터페이스(160) 또는 기타 적절한 메카니즘을 통해 시스템 버스(121)에 접속될 수 있다. 네트워크 환경에서, 컴퓨터(110)에 관하여 도시된 프로그램 모듈 또는 그 일부분은 원격 메모리 저장 장치에 저장될 수 있다. 예로서 (한정하고자 하는 것은 아님), 도 1은 메모리 장치(181)에 상주하는 원격 애플리케이션 프로그램(185)을 도시한다. 도시된 네트워크 접속은 예시적인 것이며, 컴퓨터들간의 통신 링크를 구축하는 그 외의 수단이 사용될 수 있다.

[0039] 단일 기계에 있는 복수의 컴퓨팅 환경들

[0040] 상술한 바와 같이, 2가지의 오퍼레이팅 시스템이 단일 컴퓨팅 장치 상에 나란히 실행될 수 있다는 것은 본 기술분야에 공지되어 있다. 본 발명이 다루고 있는 한가지 문제점은, 다루기 위해서 사용될 수 있는 이슈가 2개의 오퍼레이팅 시스템 사이에 일정 분리 레벨을 제공하면서 2개의 오퍼레이팅 시스템 사이에 일정 상호작용 레벨을 제공하는 방법이라는 것이다.

[0041] 도 2는 2개의 오퍼레이팅 시스템(134(1) 및 134(2))이 단일 컴퓨터(110) 상에서 실행되는 시스템을 도시한다. 몇몇 종류의 논리적 분리선(202)이 오퍼레이팅 시스템들(134(1) 및 134(2)) 사이에 존재하여, 오퍼레이팅 시스템들(134(1) 및 134(2)) 사이에서 특정량의 상호작용(204)이 허용되도록 하면서 적어도 하나의 오퍼레이팅 시스템이 나머지 오퍼레이팅 시스템에서 기원되는 이벤트들에 대해 보호되도록 한다. 도 2의 예에서, 오퍼레이팅 시스템(134(1))은 호스트 오퍼레이팅 시스템이고, 오퍼레이팅 시스템(134(2))은 상술한 바와 같이 "넥서스"와 같은 게스트 오퍼레이팅 시스템이다. 상술한 바와 같이, 오퍼레이팅 시스템(134(2))이 넥서스이면, 오퍼레이팅 시스템(134(1))의 하부구조를 빌리기 위해 오퍼레이팅 시스템(134(2))이 오퍼레이팅 시스템(134(1))과 상호작용할 수 있으면서, 오퍼레이팅 시스템(134(2))이 오퍼레이팅 시스템(134(1))에서 발생하는 액션들(부당하거나 해가 없는)로부터 그 자신을 보호하게 하고, 오퍼레이팅 시스템(134(2))이 그 거동 명세와 반대로 행동할 수 있도록, 분리선(202)을 구성하는 것이 바람직하다(그러나, 본 발명은 오퍼레이팅 시스템(134(2))이 넥서스인 경우에 한정되지 않음을 이해할 것이다).

[0042] 오퍼레이팅 시스템들(134(1) 및 134(2)) 사이의 분리선(202)은 선택적으로, 보안 모니터의 도움으로 강화될 수 있다. 보안 모니터는 양 오퍼레이팅 시스템들(134(1) 및 134(2)) 외부의 컴포넌트이고, 오퍼레이팅 시스템(134(1))으로부터 오퍼레이팅 시스템(134(2))을 보호하는데 사용될 수 있는 몇몇 보안 서비스를 제공한다. 예를 들어, 보안 모니터는 특정 하드웨어에 대한 액세스를 제어할 수 있고, 모니터의 사용을 관리할 수 있고(오퍼레이팅 시스템(134(2))에게 메모리의 어떤 일부의 유일한 사용을 제공하기 위해), 또는 안전하게 오퍼레이팅 시스템(134(1))으로부터 오퍼레이팅 시스템(134(2))으로의 데이터 통신을 용이하게 할 수 있다. 비록 보안 모니터의 사용이 요구되지는 않지만, 보안 모니터의 사용은 오퍼레이팅 시스템(134(2))가 오퍼레이팅 시스템(134(1))으로부터 어떻게 보호될 수 있는지에 대한 한가지 모델을 제시한다는 것을 유의해야 한다. 또 다른 예로서, 오퍼레이팅 시스템(134(2))은 오퍼레이팅 시스템(134(1))으로부터 그 자신을 보호하기 위해 필요한 모든 기능을 포함할 수 있다.

[0043] 도 2는 오퍼레이팅 시스템(134(1))을 "호스트"로서, 오퍼레이팅 시스템(134(2))을 "게스트"로서 도시하고 있음을 주의해야 한다. 일반적으로, 이러한 특징은, 이러한 예들에서는 오퍼레이팅 시스템(134(1))이 오퍼레이팅 시스템들(134(1) 및 134(2))(예를 들어, 장치 드라이버들, 스케쥴링 등) 모두에 의해 사용되는 특정 오퍼레이팅 시스템 히부구조를 제공하고, 오퍼레이팅 시스템(134(2))이 바람직하게는 이러한 하부 구조를 필요로 하지만 오히려 오퍼레이팅 시스템(134(1))의 하부구조 사용한다는 의미에서 "게스트"라는 사실을 나타낸다. 그러나, 오퍼레이팅 시스템을 "호스트" 또는 "게스트"로 만드는 것들의 파라미터들이 유연함을 주의해야 한다. 또한, "호스트" 및 "게스트"의 종래 개념은 호스트가 게스트의 작용들로부터 그 자신을 보호할 필요가 있다는 것을 가정함을 유의해야 한다. 그러나, 도 2의 예에서, 게스트 오퍼레이팅 시스템(134(2))은 호스트 오퍼레이팅 시스템(134(1))으로부터 그 자신을 보호할 필요가 있는 고 확실성 오퍼레이팅 시스템이라고 가정된다. 이어지는 예에서는, 구별을 위해 일반적으로 오퍼레이팅 시스템(134(1))을 "호스트"라 칭하고 오퍼레이팅 시스템(134(2))을

"게스트" 또는 "넥서스"라 칭한다. 본 명세서에 설명된 기술들은 동일한 기계에서(또는 심지어 접속된 기계들의 동일한 세트에서) 실행되는 2 이상의 오퍼레이팅 시스템들의 상호작용에 적용될 수 있음을 이해해야 한다.

[0044] 신뢰되는 에이전트에 대한 안전한 입출력

[0045] 넥서스 및 넥서스 상에서 실행되는 프로세스들("신뢰되는 에이전트들" 또는 "넥서스 에이전트들")에 안전한 입출력을 제공하기 위해 신뢰되는 사용자 인터페이스(UI) 엔진 모듈이 넥서스측에 제공된다. 신뢰되는 UI 엔진은 신뢰되는 에이전트들에게 안전한 입출력 기능을 위한 공통 자원을 제공한다.

[0046] 사용자 입력에 고 확실성을 제공하기 위해, 신뢰되는 UI 엔진은 신뢰되는 사용자 입력을 수취한다. 사용자 입력의 보안은 입력을 암호화함으로써 보안될 수 있다. 다른 실시예들에서, 사용자 입력의 보안은 디지털 서명과 같은 기타 검증 수단에 의해 검증되거나, 보안이 신뢰되는 하드웨어와 같은 다른 수단에 의해 보장된다. 신뢰되는 UI 엔진은 사용자 입력에 대한 적절한 목적지를 결정하는 신뢰되는 입력 관리자를 포함한다. 어떤 경우에는, 사용자 입력이 호스트측에서 사용되어야 한다. 이러한 경우에는, 사용자 입력은 호스트 오퍼레이팅 시스템에 대한 호스트측 상의 입력 스택에 배치될 수 있다. 그러나, 어떤 경우에는, 사용자 입력은 매우 확실하여, 신뢰되는 입력 관리자에 의해 적절한 목적지, 예를 들어 넥서스 에이전트들로 지향된다.

[0047] 출력에 고 확실성을 제공하기 위해, 신뢰되는 UI 엔진은 신뢰되는 출력 관리자에 대한 기타 프로세스들 및 에이전트들로부터의 신뢰되는 출력을 수취한다. 신뢰되는 출력 관리자는 컴퓨터 시스템(110)이 신뢰되는 출력을 넥서스에 제공하는 방법에 기초하여, 이러한 신뢰되는 출력을 수신하고 신뢰되는 출력의 표시를 장치 의존 방식으로 처리한다. 일 실시예에서, 암호화된 메모리는 전용 표시 영역들을 기술하는데 사용된다. 이러한 암호화된 메모리는 호스트측으로 전달된다. 메모리가 암호화되기 때문에, 호스트측은 메모리를 암호해독하거나 전용 표시 영역들에 대한 가짜 내용을 생성할 수 없다. 또 다른 실시예에서는, 커튼드 메모리가 사용되며, 이와 함께 그 메모리용의 내장형 비디오 어댑터가 사용된다. 신뢰되는 출력 관리자는 넥서스 에이전트들에 장치 독립 방식으로 표시 데이터를 출력하는 능력을 제공한다. 출력은 일반적으로 시각적 출력으로서 기술되지만, 신뢰되는 출력 관리자 및 본 명세서에 기재된 기술들의 사용은 임의 형태의 신뢰되는 출력(예를 들어, 오디오, 프린터, 또는 기타 출력들)에 적용될 수 있고, 본 발명은 시각적 출력에 한정되지 않음을 이해해야 한다.

[0048] 신뢰되는 UI 엔진의 안전한 입력 기능

[0049] 본 발명의 일 실시예에 따르면, 입력에 대해서는, 신뢰되는 UI 엔진은 신뢰되는 입력에 대한 액세스를 제어한다. 사용자가 고 확실성 오퍼레이팅 시스템을 포함하는 컴퓨터 시스템의 프로그램들과 상호작용하는 경우, 사용자는 (도 1의) 마우스(161) 또는 키보드(162)와 같은 사용자 입력 장치로 상호작용한다. 일 실시예에서, 사용자 입력 장치들의 일부 또는 전부는 암호화를 지원할 수 있다. 이러한 암호화된 장치들은 입력 장치 및 넥서스가 안전한 접속을 확립하도록 허용하기 때문에 하드웨어 어택을 방지할 수 있다. 암호화되지 않은 다른 입력 장치들은 존재할 수 잇지만, 암호화되지 않은 사용자 입력 장치들로부터의 입력은 단지 호스트 시스템에 대해서만 수취되고, 넥서스 또는 넥서스에서 실행되는 임의의 엔티티들에 대해서는 수취되지 않는다.

[0050] 따라서, 안전한 입력을 제공하기 위해, 호스트측 엔티티가 아니라 신뢰되는 UI 엔진은 사용자 입력의 적어도 하나의 스트림의 초기 흐름을 제어할 것이다. 도 3은 본 발명의 일 실시예에 따른 입력을 도시하는 2개의 실행 환경들을 갖는 컴퓨터 시스템 블록도이다. 도 3에 도시된 바와 같이, 신뢰되는 입력 장치(300)는 암호화된 사용자 입력 데이터를 컴퓨터 시스템(110)으로 송신한다. 암호화된 데이터는 호스트측 오퍼레이팅 시스템(134(1)) 내의 입력 스택(305)에 도착한다. 그것은 암호화되어 있기 때문에, 호스트측 오퍼레이팅 시스템(134(1))에 의해 사용될 수 없다. 따라서, 호스트측에서의 가능한 어택으로부터 안전해진다. 암호화된 데이터는 넥서스(134(2))로 전달된다.

[0051] 도 3에서, 호스트측(논리적 분리선(202)의 우측)과 넥서스측(논리적 분리선(202)의 좌측) 모두는 사용자 레벨(310) 및 커넬 레벨(320)로 더 분할된다. 이들은 오퍼레이팅 시스템의 사용자 레벨과 커넬 레벨 실행 사이의 차이에 대응한다. 넥서스(134(2))가 암호화된 사용자 입력 데이터를 수신하면, 그것은 입력된 TSP(325: trusted service provider)로 전달되고, 여기서 암호화된 사용자 입력 데이터가 암호해독된다. 몇몇 실시예에서는, 안전한 입력 장치(300)는, 컴퓨터 시스템(110)과 통신하기 위해, 예를 들어 공유된 기밀 및 검증을 사용하여 통신 채널을 확립해야 한다. 통신 채널이 안전한 입력 장치(300)에 의해 요구되면, TSP(325)는 안전한 입력 장치와의 통신 채널을 확립하고 관리한다. 또 다른 실시예에서, 사용자 입력 데이터는 검증가능하고, TSP(325)는 예를 들어 사용자 입력 데이터의 디지털 서명을 확인함으로써 사용자 입력 데이터에 대한 검증을 제공한다.

[0052] 그 후 암호해독된 사용자 입력 데이터는 신뢰되는 사용자 인터페이스(UI) 엔진(TUE)(330), 특히 신뢰되는 입력 관리자(340)로 전달된다. 사용자 입력이 호스트측으로 지향되어야 하는지, 또는 넥서스 에이전트(375)와 같은 넥서스 또는 넥서스 에이전트용으로 의도되는지를 신뢰되는 입력 관리자가 결정한다. 입력이 호스트측에서 사용되도록 정해지면, 화살표 C(390)로 도시된 바와 같이, 호스트측에서 사용되도록 입력 스택(305)에 재저장된다. 따라서, 암호해독된 사용자 입력 데이터만이 넥서스측의 결정에 의해 호스트측에 도달한다.

[0053] 일 실시예에서, 컴퓨터 시스템(110)은 윈도우, 다이얼로그 박스, 및 아이콘과 같은 그래픽 사용자 인터페이스 요소들로 이루어지는 윈도우화 인터페이스 환경을 사용자에게 제시한다. 그래픽 사용자 인터페이스 요소들은 호스트측에서의 프로세스 또는 넥서스측에서의 프로세스와 관련되거나 이에 의해 "소유될" 수 있다. 윈도우화 시스템을 사용하는 경우, 사용자의 디스플레이는 윈도우들로 이루어질 수 있으며, 그 스크린 영역들은 어플리케이션으로부터의 정보를 표시한다. 어플리케이션은 하나 이상의 윈도우를 가질 수 있다. 표시된 모든 윈도우들 중 하나의 윈도우는 포커스를 가질 수 있다. 포커스 윈도우는 예를 들어 윈도우 주위의 상이한 경계에 의해 표시될 수 있다.

[0054] 종래의 윈도우화 시스템에서, 윈도우가 포커스를 가지면, 그것은 일반적으로 사용자 입력의 오브젝트이다. 따라서, 사용자가 키보드를 이용하여 정보를 타이핑하면, 많은 경우에, 오퍼레이팅 시스템에 의해 포커스를 갖는 윈도우를 소유한 어플리케이션으로 키스트로크(keystroke) 데이터가 송신될 것이다. 몇몇 키스트로크들 및 기타 입력 액션들은 포커스를 갖는 윈도우를 소유한 어플리케이션으로 송신되지 않을 수 있다. 예를 들어, 몇몇 윈도우화 시스템들에는 모든 윈도우들을 최소화하는 키스트로크 커맨드가 있다. 이러한 커맨드는 윈도우화 시스템에 의해 처리될 것이고, 포커스를 갖는 윈도우를 소유한 어플리케이션에 송신되지 않는다. 포커스가 있는 윈도우(focused-on window)를 소유한 어플리케이션은 윈도우의 최소화 통지를 수신할 수 있지만, 사용자 키스트로크들은 포커스를 갖는 윈도우를 소유한 어플리케이션이 아니라 윈도우화 시스템들로 의도되고, 그 어플리케이션으로 송신되지 않을 것이다.

[0055] 일 실시예에서, 신뢰되는 UI 엔진은 신뢰되는 윈도우 관리자(345)를 포함한다. 이 신뢰되는 윈도우 관리자(345)는 넥서스측의 윈도우들 및 윈도우 거동을 관리한다. 신뢰되는 윈도우 관리자(345)는 신뢰되는 입력 관리자(340)과 함께 작용하여 사용자 입력 데이터가 패스되어야 하는지를 결정한다. 일 실시예에서, 이러한 결정은 적어도 부분적으로 포커스를 갖는 그래픽 사용자 요소 또는 윈도우가 넥서스의 프로세스 또는 호스트의 프로세스에 의해 소유되는지에 기초한다.

[0056] 신뢰되는 윈도우 관리자는 경계선들과 같은 윈도우 데코레이션을 유지하고 윈도우의 최소화, 최대화 및 사이즈 변경과 같은 윈도우 기능들을 위해 사용자 입력에 응답한다. 일 실시예에서, 신뢰되는 윈도우 관리자는 또한 사용자 입력 데이터의 일부 해석을 수행한다. 예를 들어, 일 실시예에서는 마우스 버튼이 눌려졌다는 것 또는 키스트로크가 수행되었다는 것을 나타내는 원래 사용자 입력 데이터가 호스트 윈도우 관리자(346)에 의해 일반적으로 수행되는 기능과 유사한, 넥서스 에이전트(375)와 같은 넥서스 에이전트용의 보다 사용가능한 형태로 해석될 것이다.

[0057] 입력이 넥서스 에이전트(375)와 같은 넥서스 에이전트용이면, 데이터는 신뢰되는 입력 관리자로부터 보정 목적지 넥서스-모드 프로세스로 전달될 것이다. 이것은 화살표 A(370)로 도시되고, 사용자 입력 데이터가 넥서스 에이전트(375)로 송신되는 것을 나타낸다. 일 실시예에서, 기타 데이터는 TOM(350)으로 또는 넥서스측의 다른 엔티티들로 전달될 수 있다. 넥서스측 그래픽 사용자 인터페이스 요소들의 표시는 신뢰되는 출력 관리자(TOM)(350)에 의해 처리되어 넥서스측 사용자 인터페이스 요소들에서의 마우스 움직임이 화살표 B(380)로 도시된 바와 같이 TOM(350)으로 전달된다.

[0058] 도 4는 본 발명의 일 실시예에 따라 보안되는 실행 환경(넥서스)의 엔티티들로부터 안전한 입력을 제공하는 방법의 흐름도이다. 단계 400에서, 사용자 입력은 사용자 입력 장치로부터 수취된다. 단계 410에서, 사용자 입력이 넥서스로 의도되는지, 예를 들어 사용자 입력이 넥서스에서 실행되는 에이전트용인지가 결정된다. 단계 420에서, 사용자 입력이 넥서스로 의도되지 않으면, 사용자 입력은 호스트로 전송된다.

[0059] 신뢰되는 UI 엔진의 안전한 출력 기능

[0060] 신뢰되는 출력 엔진의 안전한 출력 기능은 신뢰되는 출력 관리자(350)를 통해 신뢰되는 UI 엔진(330)에 의해 제공된다. 도 5는 본 발명의 일 실시예에 따른 출력을 도시하는 3개의 실행 환경들을 갖는 컴퓨터 시스템 블록도이다. 도 5의 화살표 D(500)로 도시된 바와 같이, 넥서스 에이전트(375)가 출력 장치(520)용 출력을 갖는 경우에, 그 출력은 신뢰되는 UI 엔진(330)으로 중계된다. 출력에 보안을 제공하기 위해, 화살표 E(510)로 도시된

바와 같이, 신뢰되는 출력 관리자(340)는 출력 신뢰되는 서비스 제공자(TSP)(530)와, 그리고 이에 따라 출력 장치(520)와 넥서스와의 컨택트 포인트로서 기능한다.

[0061] 상술한 바와 같이, 출력 장치(520)가 보안될 수 있는 여러가지 방식이 있다. 일 실시예에서는, 암호화가 사용된다. 출력 장치(520)가 올바르게 암호해독하는 암호화된 형태로 출력 데이터를 수신하면, 출력 장치는 그 출력 데이터를 우선시켜 디스플레이 상에서 논-넥서스 출력 데이터를 오바라이트하게 한다. 또 다른 실시예에서는, 커튼드 메모리가 사용된다. 신뢰되는 출력 관리자(350)는 출력 TSP(530)와 상호작용하여 출력 장치(520)의 보안을 유지한다.

[0062] 출력에 존재하는 보안의 형태와 관계없이, 신뢰되는 출력 관리자(350)는 신뢰되는 윈도우 관리자(345)를 포함하는 컴퓨터 시스템(110)에서, 신뢰되는 윈도우 관리자(345)로부터의 임의의 출력 정보를 처리하는 것을 포함해서, 넥서스측과 출력 장치(520) 사이의 모든 상호작용들을 관리한다. 넥서스 에이전트(375)와 같은 넥서스 에이전트들에 의해 소유되는 윈도우로부터의 출력 정보는 신뢰되는 출력 관리자(350)에 의해 처리된다. 또한, 임의의 다른 넥서스 그래픽 사용자 인터페이스 요소들이 존재하면, 이러한 요소들의 표시가 신뢰되는 출력 관리자에 의해 처리된다. 예를 들어, 마우스가 그러한 넥서스 윈도우 위를 움직이면, 보안 문제를 유발하지 않고 호스트측이 마우스를 제어하도록 하는 것은 어려울 수 있다. 따라서, 마우스가 넥서스 윈도우 위에 있는 경우의 마우스 커서의 표시는 넥서스측에서 처리될 수 있다. 모든 넥서스 윈도우 및 기타 넥서스 그래픽 사용자 인터페이스 요소들의 출력 정보는 표시를 위해 신뢰되는 출력 관리자에 의해 합성될 수 있다. 부가적으로, 호스트 윈도우 관리자(346)와의 임의의 상호작용은 신뢰되는 UI 엔진(330)에 의해 처리된다.

[0063] 신뢰되는 UI 엔진의 안전한 출력 기능

[0064] 일반적으로, 다이얼로그 프레임워크 서비스들은 프로세서를 위해 제공될 수 있다. 이러한 서비스들은 원하는 그래픽 사용자 인터페이스 요소들을 기술하는 파일을 수취하여 표시한다. 예를 들어, 파일은 특정 질문, 응답 선택들, 및 대화 박스를 제시할 때 사용하는 언어를 포함하여, 사용자에게 제시하기 위한 대화 박스를 기술하는 정보를 가진 XML 파일일 수 있다. 대화 프레임워크 서비스는 올바른 대화 박스를 렌더링하는 정보를 포함한다. 그러나, 호스트측의 대화 프레임워크 서비스에 의존하는 것은 보안 문제를 가져올 것이다. 따라서 일 실시예에서는, 신뢰되는 UI 엔진(330) 내에 신뢰되는 렌더링 인터페이스가 제공된다. 이 신뢰되는 렌더링 인터페이스는 넥서스 에이전트(375)와 같은 넥서스 에이전트들에 완전한 대화 프레임워크 서비스를 제공한다. 신뢰되는 UI 엔진(330)의 일부로서, 신뢰되는 렌더링 인터페이스는 표시를 위한 그래픽 사용자 인터페이스 요소, 및 신뢰되는 렌더링 인터페이스가 요구된 그래픽 사용자 인터페이스를 렌더링하고 그것을 관리하는 방법을 특정하는 정보를 수신하고, 그 내부의 활성 그래픽 사용자 인터페이스 요소들에서 사용자 이벤트가 발생한 경우에 넥서스 에이전트(375)에게 경고를 보낸다. 이러한 방식으로, 신뢰되는 UI 엔진(330)은 호스트측의 국부화 서비스들의 사용을 요구하지 않고 넥서스 에이전트(375)가 그래픽 사용자 인터페이스에 요구하는 국부화 서비스들의 일부를 제공한다. 예를 들어, 신뢰되는 UI 엔진(330)은 다중 언어 텍스트 서비스를 제공할 수 있다.

[0065] 일 실시예에서, 신뢰되는 UI 엔진(330)에는 직접 렌더링 인터페이스가 제공된다. 이러한 직접 렌더링 인터페이스는 신뢰되는 렌더링 인터페이스에 부가하여 제공될 수 있고, 또는 신뢰되는 렌더링 인터페이스가 존재하지 않는 곳에도 존재할 수 있다. 직접 렌더링 인터페이스는 그 자신의 렌더링을 수행하고자 하는 넥서스 에이전트(375)와 같은 넥서스 에이전트에 베어-본(bare-bones) 메카니즘을 제공한다. 넥서스 에이전트(375)는 신뢰되는 UI 엔진(330)에 의존하기 보다는 그 자신의 스크린 비트맵을 산출하여 그것을 명세로부터 렌더링한다. 신뢰되는 UI 엔진은 신뢰되는 입력 관리자(340)를 통해 사용자 입력 액션들의 통지를 송신한다.

[0066] 도 6은 본 발명의 일 실시예에 따른 보안되는 실행 환경에 있는 엔티티들로부터의 안전한 출력을 제공하기 위한 방법의 흐름도이다. 단계 600에서, 출력은 넥서스와 같은 보안되는 실행 환경에 있는 특정 소스 엔티티로부터 수취된다. 단계 610에서, 출력은 출력 장치로 안전하게 전송된다.

[0067] 결론

[0068] 상기한 예들은 단지 예시의 목적으로 제공되었고 본 발명을 제한하는 것으로 해석되지 않음을 주의해야 한다. 본 발명이 각종 실시예들을 참조하여 설명되었지만, 본 명세서에서 사용된 단어들은 한정의 단어들이 아니라 설명 및 예시의 단어들임을 이해해야 한다. 또한, 본 발명이 특정 수단, 재료 및 실시예들을 참조하여 설명되었지만, 본 발명은 본 명세서에 개시된 세부사항들에 한정되도록 의도되지 않고, 본 발명은 모든 기능적으로 등가인 구조들, 방법들 및 사용으로 확장되며, 이러한 것은 첨부된 특허청구범위 내에 있다. 본 명세서의 교시들의 이점을 가지는 본 기술 분야의 당업자들은 각종 변경을 행할 수 있고, 특징에 있어서 본 발명의 범위 및 정신으

로부터 벗어나지 않고 변화가 이루어질 수 있다.

발명의 효과

[0069] 본 발명에 의하면, 복수의 실행 환경들 중에서 보다 높은 확실성 실행 환경의 신뢰되는 에이전트에 대한 데이터의 입출력 보전성을 가능케 하는 기술들이 제공된다.

도면의 간단한 설명

[0001] 도 1은 본 발명의 특징들이 구현될 수 있는 예시적인 컴퓨팅 환경의 블록도.

[0002] 도 2는 서로 몇몇 상호작용을 유지하고 서로 몇몇 분리를 유지하는 2가지의 예시적인 실행 환경들에 대한 블록도.

[0003] 도 3은 본 발명의 일 실시예에 따른 입력을 도시하는 2가지의 실행 환경들을 갖는 컴퓨터 시스템의 블록도.

[0004] 도 4는 본 발명의 일 실시예에 따른 보안되는 실행 환경의 엔티티들로부터 보안 입력을 제공하는 방법의 흐름도.

[0005] 도 5는 본 발명의 일 실시예에 따른 출력을 도시하는 2가지의 실행 환경들을 갖는 컴퓨터 시스템의 블록도.

[0006] 도 6은 본 발명의 일 실시예에 따른 보안되는 실행 환경의 엔티티들로부터 보안 출력을 제공하는 방법의 흐름도.

[0007] <도면의 주요 부분에 대한 부호의 설명>

[0008] 520 : 출력 장치

[0009] 110 : 컴퓨터

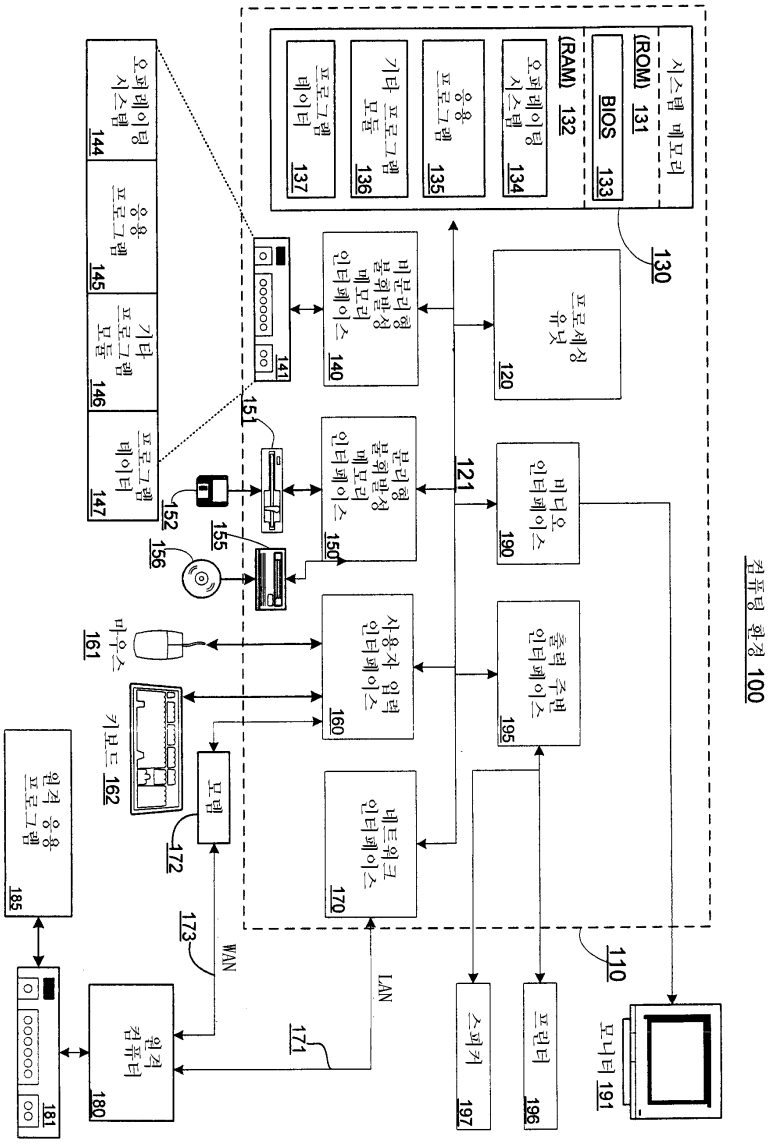
[0010] 375 : 넥서스 에이전트

[0011] 346 : 호스트 윈도우 관리자

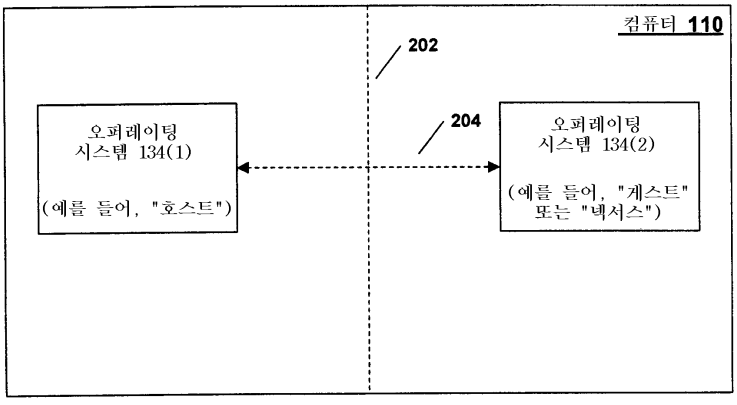
[0012] 310 : 사용자 레벨

도면

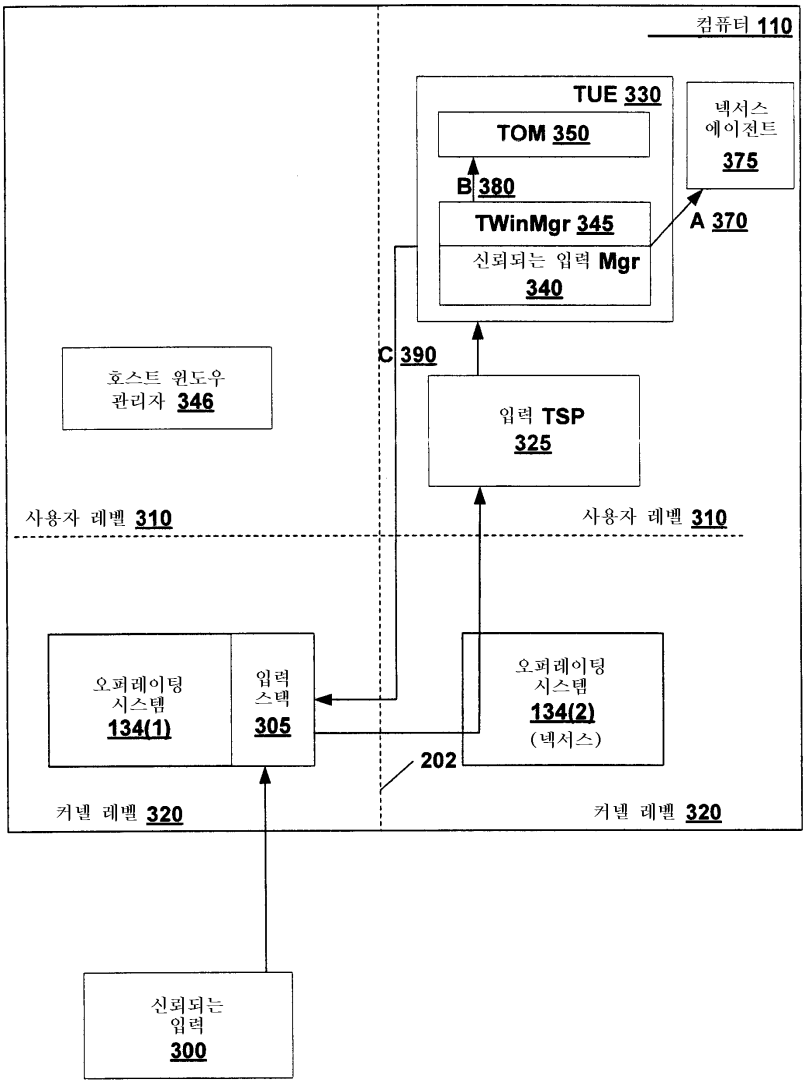
도면1


컴퓨팅 환경 100
시스템 메모리
(ROM) 131
BIOS 133
(RAM) 132
오퍼레이팅 시스템 134
응용 프로그램 135
기타 프로그램 모듈 136
프로그램 데이터 137
130
프로세싱 유닛 120
121
비분리형 불휘발성 메모리 인터페이스 140
분리형 불휘발성 메모리 인터페이스 150
비디오 인터페이스 190
출력 주변 인터페이스 195
사용자 입력 인터페이스 160
네트워크 인터페이스 170
모뎀 172
WAN
LAN
171
173
151
152
155
156
141
오퍼레이팅 시스템 144
응용 프로그램 145
기타 프로그램 모듈 146
프로그램 데이터 147
마우스 161
키보드 162
원격 응용 프로그램 185
원격 컴퓨터 180
181
스피커 197
프린터 196
모니터 191
110

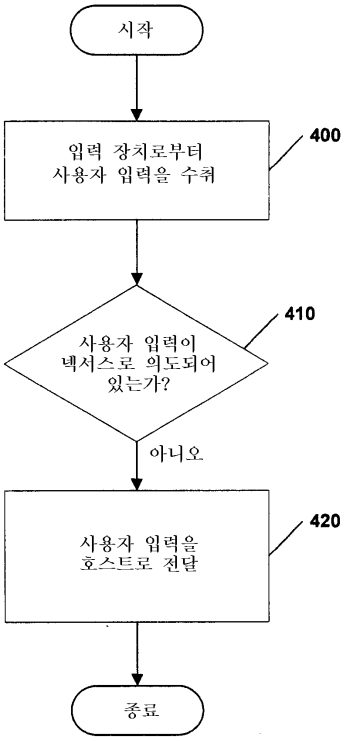
도면2


컴퓨터 110
202
204
오퍼레이팅 시스템 134(1)
(예를 들어, "호스트")
오퍼레이팅 시스템 134(2)
(예를 들어, "게스트" 또는 "넥서스")

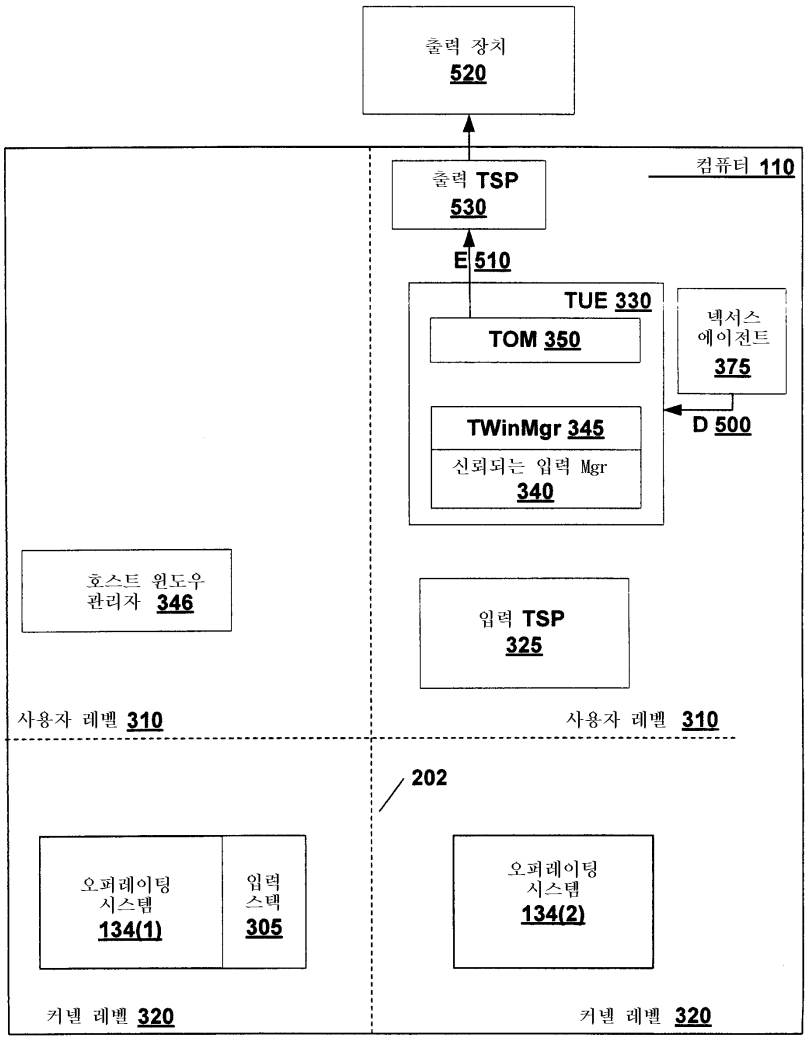
도면3


컴퓨터 110
TUE 330
TOM 350
B 380
TWinMgr 345
신뢰되는 입력 Mgr
340
넥서스
에이전트
375
A 370
C 390
호스트 윈도우
관리자 346
입력 TSP
325
사용자 레벨 310
사용자 레벨 310
오퍼레이팅
시스템
134(1)
입력
스택
305
오퍼레이팅
시스템
134(2)
(넥서스)
202
커널 레벨 320
커널 레벨 320
신뢰되는
입력
300

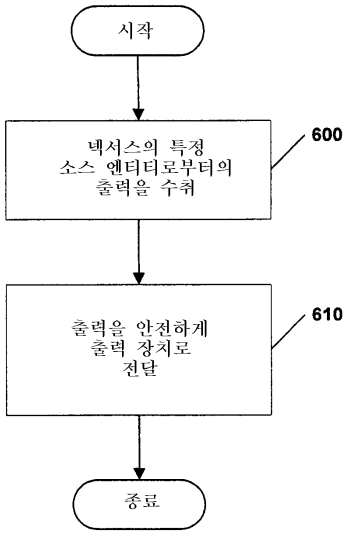
도면4


시작
입력 장치로부터
사용자 입력을 수취
400
사용자 입력이
넥서스로 의도되어
있는가?
410
아니오
사용자 입력을
호스트로 전달
420
종료

도면5


출력 장치
520
컴퓨터 110
출력 TSP
530
E 510
TUE 330
TOM 350
넥서스
에이전트
375
D 500
TWinMgr 345
신뢰되는 입력 Mgr
340
호스트 윈도우
관리자 346
입력 TSP
325
사용자 레벨 310
사용자 레벨 310
202
오퍼레이팅
시스템
134(1)
입력
스택
305
오퍼레이팅
시스템
134(2)
커널 레벨 320
커널 레벨 320

도면6


시작
넥서스의 특정
소스 엔티티로부터의
출력을 수취
600
출력을 안전하게
출력 장치로
전달
610
종료