(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0077846
(43) 공개일자 2013년07월09일

(51) 국제특허분류(Int. Cl.)
G06F 11/28 (2006.01) *G06F 9/44* (2006.01)
(21) 출원번호 10-2012-7034165
(22) 출원일자(국제) 2011년05월09일
심사청구일자 2012년12월27일
(85) 번역문제출일자 2012년12월27일
(86) 국제출원번호 PCT/KR2011/003436
(87) 국제공개번호 WO 2012/153879
국제공개일자 2012년11월15일

(71) 출원인
현대자동차주식회사
서울특별시 서초구 헌릉로 12 (양재동)
기아자동차주식회사
서울특별시 서초구 헌릉로 12 (양재동)
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
최병주
서울특별시 강남구 압구정동 현대아파트 74-1305
서주영
서울특별시 송파구 문정동 올림픽훼밀리타운 105-304
(뒷면에 계속)
(74) 대리인
특허법인태평양

전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **예외처리 테스트 장치 및 방법**

(57) 요 약

본 발명은 예외처리 테스트 장치에 관한 것으로서, 결함 모델 및 디바이스 매니저로부터 획득한 정보를 기초로 변형된 디바이스 드라이버를 생성하는 생성 모듈, 상기 변형된 디바이스 드라이버를 이용하여 상기 디바이스 드라이버를 후킹하는 후킹 모듈, 상기 어플리케이션이 동작하는 동안, 후킹된 상기 변형된 디바이스 드라이버로부터 상기 어플리케이션으로 반환되는 테스트 정보를 수집하는 스캐닝 모듈 및 상기 스캐닝 모듈에 의해 수집된 테스트 정보를 분석하는 분석 모듈을 포함한다.

(72) 발명자

양승완

경기도 군포시 광정동 삼성장미아파트 1137-2202

김영수

경기도 용인시 기흥구 신갈동 697번지 녹원마을 새천년그린빌 509동 503호

오정석

경기도 광명시 광명1동 70-13, 2층

권해영

경기도 용인시 기흥구 보정동 현대아이파크 1차 아파트 202-601

장승연

경기도 고양시 덕양구 화정2동 별빛마을 건영아파트 1004동 1101호

특허청구의 범위

청구항 1

디바이스 드라이버, 디바이스 매니저 및 어플리케이션을 포함하는 시스템에 대한 예외처리 테스트 장치에 있어서,

결함 모델 및 디바이스 매니저로부터 획득한 정보를 기초로 변형된 디바이스 드라이버를 생성하는 생성 모듈;

상기 변형된 디바이스 드라이버를 이용하여 상기 디바이스 드라이버를 후킹하는 후킹 모듈;

상기 어플리케이션이 동작하는 동안, 후킹된 상기 변형된 디바이스 드라이버로부터 상기 어플리케이션으로 반환되는 테스트 정보를 수집하는 스캐닝 모듈; 및

상기 스캐닝 모듈에 의해 수집된 테스트 정보를 분석하는 분석 모듈을 포함하는 예외처리 테스트 장치.

청구항 2

제1항에 있어서,

상기 결함 모델은 어플리케이션과 디바이스 사이의 연결, 디바이스 오픈, 연결해제, 디바이스 닫기, 널(NULL) 데이터, 데이터 오류, 디바이스 전원, 및 처리지연 중 적어도 하나인 것을 특징으로 하는 예외처리 테스트 장치.

청구항 3

제1항에 있어서,

상기 테스트 정보는 상기 변형된 디바이스 드라이버의 오퍼레이션 반환 값, 상기 디바이스 매니저에서 발생하는 예외 코드 값, 상기 어플리케이션의 반환 값, 및 시스템 이상 코드 중 적어도 하나인 것을 특징으로 하는 예외처리 테스트 장치.

청구항 4

제1항에 있어서,

상기 테스트 정보가 결함이 발생되었음을 의미하는 경우, 상기 어플리케이션은 상기 발생된 결함에 따라 예외처리를 수행하는 것을 특징으로 하는 예외처리 테스트 장치.

청구항 5

제4항에 있어서,

상기 예외처리가 정상적으로 수행되지 않는 경우, 상기 분석 모듈은 상기 테스트 정보를 전체 시스템 다운, 일부 프로세스 다운, 일부 프로세스의 비정상적인 종료, 일부 프로세스의 비정상적인 동작, 비정상 리턴하고 에러코드 없음, 및 정상 리턴하고 부정확한 에러코드 반환 중 적어도 하나로 분석하는 것을 특징으로 하는 예외처리 테스트 장치.

청구항 6

제1항에 있어서,

상기 결함 모델은 테스트 대상이 되는 디바이스 및 디바이스 드라이버에 따라 변경되는 것을 특징으로 하는 예외처리 테스트 장치.

청구항 7

디바이스 드라이버, 디바이스 매니저 및 어플리케이션을 포함하는 시스템에 대한 예외처리 테스트 방법에 있어서,

결함 모델 및 상기 디바이스 매니저로부터 획득한 정보를 기초로 변형된 디바이스 드라이버를 생성하는 생성 단계;

상기 변형된 디바이스 드라이버를 이용하여 상기 디바이스 드라이버를 후킹하는 후킹 단계;

상기 어플리케이션이 동작하는 동안, 후킹된 상기 변형된 디바이스 드라이버로부터 상기 어플리케이션으로 반환되는 테스트 정보를 수집하는 스캐닝 단계; 및

수집된 테스트 정보를 분석하는 분석 단계를 포함하는 예외처리 테스트 방법.

청구항 8

제7항에 있어서,

상기 결함 모델은 어플리케이션과 디바이스 사이의 연결, 디바이스 오픈, 연결해제, 디바이스 닫기, 널(NULL) 데이터, 데이터 오류, 디바이스 전원, 및 처리지연 중 적어도 하나인 것을 특징으로 하는 예외처리 테스트 방법.

청구항 9

제7항에 있어서,

상기 테스트 정보는 상기 변형된 디바이스 드라이버의 오퍼레이션 반환 값, 상기 디바이스 매니저에서 발생하는 예외 코드 값, 상기 어플리케이션의 반환 값, 및 시스템 이상 코드 중 적어도 하나인 것을 특징으로 하는 예외처리 테스트 방법.

청구항 10

제7항에 있어서,

상기 테스트 정보가 결함이 발생되었음을 의미하는 경우, 상기 어플리케이션은 상기 발생된 결함에 따라 예외처리를 수행하는 것을 특징으로 하는 예외처리 테스트 방법.

청구항 11

제10항에 있어서,

상기 예외처리가 정상적으로 수행되지 않는 경우, 상기 분석 단계는 상기 테스트 정보를 전체 시스템 다운, 일부 프로세스 다운, 일부 프로세스의 비정상적인 종료, 일부 프로세스의 비정상적인 동작, 비정상 리턴하고 에러코드 없음, 및 정상 리턴하고 부정확한 에러코드 반환 중 적어도 하나로 분석하는 것을 특징으로 하는 예외처리 테스트 방법.

청구항 12

제7항에 있어서,

상기 결함 모델은 테스트 대상이 되는 디바이스 및 디바이스 드라이버에 따라 변경되는 것을 특징으로 하는 예외처리 테스트 방법.

명 세 서

기 술 분 야

[0001] 본 발명은 예외처리 테스트 장치 및 방법에 관한 것으로, 보다 상세하게는 변형된 디바이스 드라이버를 이용하여 결함을 발생시키고, 상기 결함에 대한 시스템의 예외처리 여부를 확인하는 예외처리 테스트 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 다양한 소프트웨어 및 하드웨어를 포함하는 임베디드 시스템에 있어서, 시스템의 운영환경에서 상호작용(interaction)하는 동작이 중요하다. 따라서, 소프트웨어만으로 구성된 테스트를 위한 가상 데이터가 아닌, 시스템이 실제 운영되면서 얻게 되는 실제 데이터(live environment with real data)를 이용하는 동작이 중요하며, 시스템을 테스트 하고자 하는 경우에도, 실제 운용 데이터를 이용하여 시스템 테스트를 수행해야 잠재적 결함 또는 실제 운용상 발생하는 운용상 결함을 효과적으로 찾아낼 수 있는 방안이 요구된다.

[0003] 또한, 시스템을 구성하는 소프트웨어 각각을 별도로 독립적으로 테스트하려면 연결되는 상대 컴포넌트를 대체하기 위한 장비를 갖추어야 하는 등 테스트 자원과 비용이 많이 필요하기 때문에, 시스템에 포함된 모든 프로세스를 실시간으로 테스트할 수 있고, 메모리 공간 오버헤드 또는 해당 코드 수행으로 인한 성능 오버헤드를 최소화하는 테스트 장치가 요구된다.

발명의 내용

해결하려는 과제

[0004] 본 발명은 후킹 기술을 이용하여 변형된 디바이스 드라이버를 시스템에 삽입하여, 상기 변형된 디바이스 드라이버에 의해 발생하는 결함을 확인하고, 확인된 결함에 따른 예외처리 여부를 확인함으로써, 실제 운영되는 시스템에 대한 테스트가 가능하고, 테스트 자원 오버헤드를 최소화할 수 있는 예외처리 테스트 장치 및 방법을 제공하는데 그 목적이 있다.

과제의 해결 수단

[0005] 본 발명은 예외처리 테스트 장치에 관한 것으로서, 결함 모델 및 디바이스 매니저로부터 획득한 정보를 기초로 변형된 디바이스 드라이버를 생성하는 생성 모듈, 상기 변형된 디바이스 드라이버를 이용하여 상기 디바이스 드라이버를 후킹하는 후킹 모듈, 어플리케이션이 동작하는 동안, 후킹된 상기 변형된 디바이스 드라이버로부터 상기 어플리케이션으로 반환되는 테스트 정보를 수집하는 스캐닝 모듈, 및 스캐닝 모듈에 의해 수집된 테스트 정보를 분석하는 분석 모듈을 포함한다.

[0006] 본 발명은 예외처리 테스트 방법에 관한 것으로서, 결함 모델 및 디바이스 매니저로부터 획득한 정보를 기초로 변형된 디바이스 드라이버를 생성하는 생성 단계, 상기 변형된 디바이스 드라이버를 이용하여 상기 디바이스 드라이버를 후킹하는 후킹 단계, 상기 어플리케이션이 동작하는 동안, 후킹된 상기 변형된 디바이스 드라이버로부터 상기 어플리케이션으로 반환되는 테스트 정보를 수집하는 스캐닝 단계, 및 수집된 테스트 정보를 분석하는 분석 단계를 포함한다.

발명의 효과

[0007] 본 발명에 따르면, 소스 코드가 아닌 실제 운영되는 코드를 대상으로 하므로, 시스템이 실제로 운용되는 동안에도 디바이스 매니저를 이용한 예외처리 테스트를 수행할 수 있다.

[0008] 또한, 각각의 프로세스에 대한 개별적인 코드 삽입이 아닌 소정의 메모리 공간의 프로세스 컨텍스트 정보를 활용한 후킹 방법을 이용하므로, 시스템 오버헤드를 최소화할 수 있다.

[0009] 또한, 프로세스 사이의 공유 자원이 제한된 경우에도, 실시간으로 테스트 정보를 수집하여 테스트 결과를 확인할 수 있다.

도면의 간단한 설명

[0010] 도 1은 본 발명의 일실시예에 따른 예외처리 테스트 장치가 이용되는 전체 시스템을 나타내는 도면이다.

도 2는 본 발명의 일실시예에 따른 예외처리 테스트 장치를 나타내는 구성도이다.

도 3은 본 발명의 일실시예에 따른 예외처리 테스트 방법을 나타내는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0011] 본 발명은 시스템이 실제 운영되는 환경에서의 테스트이며, 시스템이 시작되어 종료되기까지의 동적 상황에서 시스템을 테스트할 수 있는 장치 및 방법에 관한 것이다. 본 발명은 코드 삽입 기술의 일종인 후킹을 이용하여, 시스템 내에서 동작하는 모든 프로세스에 대한 테스트를 수행할 수 있다. 보다 상세하게는 시스템 내에서 동작하는 디바이스로부터 발생할 수 있는 결함 유형 및 상기 결함에 대한 예외처리 테스트를 수행할 수 있다.

[0012] 도 1은 본 발명의 일실시예에 따른 예외처리 테스트 장치가 이용되는 전체 시스템을 나타내는 도면이다.

[0013] 도 1에 도시된 바와 같이, 전체 시스템은 예외처리 테스트 장치(100), 시스템(200), 입력 장치(300), 저장 장치(400) 및 출력 장치(500)를 포함한다.

[0014] 예외처리 테스트 장치(100)는 시스템(200)에 포함된 어플리케이션과 디바이스 사이의 통신과정에서 발생하는 결함 유형 및 상기 결함에 대한 예외처리를 테스트 하기 위한 장치이다. 예외처리 테스트 장치(100)는 시스템에 포함된 디바이스를 관리하는 디바이스 매니저를 이용하여 변형된 디바이스 드라이버를 생성하고, 상기 변형된 디바이스 드라이버를 이용하여 결함을 생성함으로써, 시스템을 테스트할 수 있다. 상기 예외처리 테스트 장치(100)는 도 1에 도시된 바와 같이 시스템(200)과 독립적으로 구비될 수도 있으며, 시스템(200) 내부에 포함되는 형태로 구현될 수도 있다.

[0015] 상기 시스템(200)은 멀티 프로세스가 통신하는 모든 임베디드 시스템 및 컴퓨터시스템으로서, 예를 들어, 인포테인먼트 시스템, 멀티미디어시스템, 유무선네트워크시스템 등이 될 수 있다. 예를 들어, 시스템(200)이 차량 인포테인먼트 시스템인 경우, 시스템(200)은 AVN(Audio Video Navigation) 기능을 중심으로 TDMB(Terrestrial Digital Multimedia Broadcasting), TPEG(Transport Protocol Expert Group), IPOD, USB, 블루투스(Bluetooth), VR (Voice Recognition), PGS (Parking Guidance System), RDS (Radio Data System), VCDC (Vehicle CDC), DIS (Driver Information System), RSE (Rear Seat Entertainment), HEV (Hybrid Electric Vehicle) display, AHU (Air Handling Unit) display 등을 포함한다.

[0016] 또한, 상기 시스템(200)은 디바이스, 상기 디바이스와 어플리케이션 사이의 데이터 송수신을 위한 디바이스 드라이버, 상기 디바이스를 관리하기 위한 디바이스 매니저 및 상기 디바이스 매니저와 연결되어 디바이스를 이용하여 소정의 프로그램을 실행시키는 어플리케이션을 포함한다. 따라서, 예외처리 테스트 장치(100)는 디바이스 매니저를 이용하여 현재 활성화된 디바이스 드라이버를 확인하고, 상기 디바이스 드라이버에 대응하는 변형된 디바이스 드라이버를 생성하고, 상기 변형된 디바이스 드라이버를 이용하여 디바이스 드라이버를 후킹하므로써, 예외처리 테스트를 실행할 수 있다. 여기서, 디바이스는 USB, 멀티미디어 플레이어 등의 하드웨어 디바이스로서, 어플리케이션이 동작하는 장치와 물리적으로 분리된 장치일 수 있다. 본 발명의 시스템(200)은 물리적으로 분리된 디바이스 및 어플리케이션이 동작하는 장치를 포함하는 개념이다.

[0017] 입력 장치(300)는 예외처리 테스트 장치(100) 및 시스템(200)을 구동시키기 위한 사용자 입력 장치이다. 입력 장치(300)는 키보드 또는 터치 스크린으로 구현될 수 있으며, 입력 장치(300)를 통해 입력되는 신호에 따라 시스템(200)에 포함되는 어플리케이션이 실행되며, 예외처리 테스트 장치(100)에 의해 변형된 디바이스 드라이버가 후킹된 시스템(200)이 동작한다.

[0018] 저장 장치(400)는 예외처리 테스트 장치(100)에 의해 수집된 정보를 저장한다. 상기 예외처리 테스트 장치(100)는 저장 장치(400)와 하나의 장치로 구현되어, 테스트 정보를 수집 및 분석함과 동시에 수집된 테스트 정보를 저장할 수 있다.

[0019] 출력 장치(500)는 예외처리 테스트 장치(100)에 의해 수집된 테스트 정보 또는 분석된 테스트 결과를 이미지 또는 음성으로 출력하는 장치이다.

[0020] 도 2는 본 발명의 일실시예에 따른 예외처리 테스트 장치를 나타내는 구성도이다.

[0021] 도 2에 도시된 바와 같이, 예외처리 테스트 장치(100)는 생성 모듈(101), 후킹 모듈(103), 스캐닝 모듈(105) 및 분석 모듈(107)을 포함한다. 예외처리 테스트 장치(100)와 연결된 시스템(200)은 디바이스(미도시), 상기 디바이스와 어플리케이션 사이의 데이터 송수신을 위한 디바이스 드라이버(미도시), 상기 디바이스를 관리하는 디바이스 매니저(201) 및 상기 디바이스 매니저(201)와 연결되어 상기 디바이스를 통해 소정의 기능을 구현하기 위한 어플리케이션(202)을 포함한다. 예외처리 테스트 장치(100)는 둘 이상의 디바이스(미도시) 및 어플리케이션(202)에 대한 예외처리 테스트를 실행할 수 있다.

[0022] 디바이스 매니저(201)는 시스템(200)에 포함된 모든 디바이스 드라이버에 대한 런-타임 실행정보를 갖는다. 여기서, 런-타임 실행정보란, 시스템(200)이 동작하는 동안 모든 디바이스와 프로세스 사이의 실행정보를 의미한다. 즉, 디바이스 매니저(201)는 시스템(200) 내에서 활성화된 디바이스 이름 및 디바이스 드라이버, 각 디바이스 드라이버에서 이용되는 오퍼레이션 및 그 오퍼레이션의 시작 주소, 디바이스 드라이버가 로딩된 메모리 주소, 메모리 공간 등에 대한 정보를 관리한다. 따라서, 모든 시스템 내에서 상기 언급된 정보를 포함하는 데이터구조 또는 장치는 본 발명에서 의미하는 디바이스 매니저(201)에 해당된다.

[0023] 생성 모듈(101)은 디바이스 매니저(201)를 이용하여 시스템(200)에서 현재 동작하고 있는 디바이스 드라이버에 대한 정보를 수집하고, 결함 모델 및 수집된 디바이스 드라이버에 대한 정보에 따라 변형된 디바이스 드라이버를 생성한다. 상기 결함 모델은 하기 도 4에서 상세하게 설명한다.

[0024] 상기 변형된 디바이스 드라이버는 디바이스와 송수신되는 데이터를 변형시켜 시스템 동작의 결함을 발생시킨다. 즉, 변형된 디바이스 드라이버는 각 오퍼레이션에서 이용되는 프로토콜 데이터를 디바이스에 따라 분석하고, 디바이스를 통과하는 데이터를 변형 규칙에 따라 변형 시킴으로써, 변형된 데이터를 어플리케이션과 디바이스 사이의 통신과정에서 이용하고 시스템 동작의 결함을 발생시킨다. 변형 규칙의 일실시예로서, 디바이스와 송수신되는 데이터가 손실되도록 하는 변형 규칙, 디바이스와 송수신되는 데이터에 유효 범위의 경계에 해당하는 값이 추가되도록 하는 변형 규칙, 데이터 전송 장애로 인해 데이터 값이 뒤섞이도록 하는 변형 규칙 등이 있다.

[0025] 입력 장치(300)에 의해 테스트 대상이 먼저 선택된 후에, 테스트 대상이 되는 디바이스 드라이버에 대한 정보가 수집될 수 있고, 현재 동작하고 있는 디바이스 드라이버에 대한 정보를 수집한 후 테스트 대상이 되는 디바이스 드라이버가 선택될 수 있다.

[0026] 후킹 모듈(103)은 원본 디바이스 드라이버를 상기 생성 모듈(101)에 의해 생성된 변형된 디바이스 드라이버로 후킹한다. 즉, 후킹 모듈(103)은 생성 모듈(101)에 의해 수집된 정보를 이용하여 원본 디바이스 드라이버를 후킹할 수 있다. 예를 들어, 디바이스 매니저(201)에 의해 관리되는 정보 중 테스트 대상이 되는 디바이스 드라이버의 오퍼레이션 시작 주소를 변형된 디바이스 드라이버의 오퍼레이션 시작 주소로 바꿈으로써, 원본 디바이스 드라이버를 후킹할 수 있다. 따라서, 시스템이 동작하는 동안 원본 디바이스 드라이버가 호출되는 대신 변형된 디바이스 드라이버가 호출된다.

[0027] 스캐닝 모듈(105)은 디바이스와 어플리케이션이 동작하는 동안, 변형된 디바이스 드라이버에 의해 실행되어 어플리케이션으로 반환되는 테스트 정보를 수집한다. 예를 들어, 디바이스 드라이버의 오퍼레이션 반환 값, 디바이스 매니저(201)에서 발생하는 예외 코드 값, 어플리케이션의 반환 값 또는 시스템 이상 코드 등을 수집할 수 있다. 또한, 스캐닝 모듈(105)은 어플리케이션(202)이 동작하는 동안, 결함 판정에 필요한 시스템 동적 상태 정보, 예를 들어, 함수 파라미터 및 반환 값, 성능 측정값 등을 수집할 수 있다.

[0028] 즉, 스캐닝 모듈(105)은 변형된 디바이스 드라이버에 의해 변형된 데이터가 이용됨으로써 발생하는 시스템 동작의 결함 유형을 수집하고, 상기 결함에 대한 시스템의 예외처리 결과를 확인한다.

[0029] 분석 모듈(107)은 상기 스캐닝 모듈(105)에 의해 수집된 예외 및 예외처리 결과를 분석한다. 즉, 수집된 정보를 기초로 하여 어플리케이션이 예외처리를 제대로 하는지 못하는지 판단된다. 예를 들어, 예외처리 결과가 PASS인 경우, 시스템이 발생한 예외에 대해 적합한 예외 코드를 출력하고, 적절한 처리를 수행했다고 분석되고, 예외처리 결과가 FAIL인 경우, 시스템이 발생한 예외에 대해 적절히 처리하지 못한 것으로 분석된다.

[0030] 상기 예외처리 결과가 FAIL인 경우는 전체 시스템 다운(Catastrophic), 일부 프로세스 다운(Restart), 일부 프로세스의 비정상적인 종료(Abort), 일부 프로세스의 비정상적인 동작(Abnormal), 비정상 리턴하고 에러코드 없음(Silent), 및 정상 리턴하고 부정확한 에러코드 반환(Hindering)으로 분류될 수 있다.

[0031] 도 3은 본 발명의 일실시예에 따른 예외처리 테스트 방법을 나타내는 흐름도이다.

[0032] 도 3은 어플리케이션과 디바이스 사이의 통신과정에서 발생하는 결함 및 상기 결함에 대한 예외처리 테스트를 수행하기 위한 예외처리 테스트 방법을 나타낸다. 먼저, 결함 모델이 정의된다(S110). 여기서, 결함 모델이란, 결함 유형에 따라 테스트되어야 하는 디바이스 및 디바이스 드라이버의 결함 발견/복구를 위한 정보를 의미한다. 결함 모델은 테스트 대상이 되는 디바이스 및 디바이스 드라이버에 따라 변경될 수 있다. 정의된 결함 모델에 따라, 시스템(200)이 정상적으로 동작하는 동안 테스트 정보를 수집할 수 있는 변형된 디바이스 드라이버가 생성될 수 있다.

[0033] 여기서, 어플리케이션과 디바이스 사이의 통신과정에서 발생하는 결함 유형은 어플리케이션과 디바이스 사이의 연결(CONNECTION), 디바이스 오픈(OPEN), 연결해제(DISCONNECTION), 디바이스 닫기(CLOSE), 데이터 입출력과 관련하여 널 데이터(NULL DATA)와 데이터 오류(ILLEGAL DATA)와 처리지연(TIMING), 또는 디바이스 전원(POWER)과 관련된 결함을 발생시키는 코드이다. 상기 결함 유형은 이하 보다 상세하게 설명한다.

[0034] 다음, S110 단계에서 정의된 결함 모델 및 시스템(200)에서 현재 동작하고 있는 디바이스 드라이버에 대한 정보를 기초로 변형된 디바이스 드라이버가 생성된다. (S120).

[0035] 다음, 원본 디바이스 드라이버가 상기 변형된 디바이스 드라이버로 후킹된다(S130). 즉, 원본 디바이스 드라이버가 호출되는 시점에서, 월본 디바이스 드라이버 대신 변형된 디바이스 드라이버가 호출된다.

[0036] 다음, 상기 디바이스 매니저(201)와 연결된 어플리케이션(202)이 구동되고, 상기 어플리케이션(202)이 동작함에

따라, 디바이스 매니저(201) 및 디바이스(미도시)가 함께 동작한다(S140).

[0037] 어플리케이션(202)이 동작하는 동안, 변형된 디바이스 드라이버에 의해 실행된 결과, 즉, 시스템 동작에서 발생한 결함 및 상기 결함에 대한 시스템의 예외처리 결과가 수집된다(S150). S150 단계에서 수집되는 테스트 정보는 현재 콜-스택 정보, 디바이스 드라이버 정보 및 프로세스 정보를 포함한다. 콜-스택 정보를 이용하여 어떤 순서로 흐름이 흐르다가 문제가 발생했는지가 확인될 수 있다. 디바이스 드라이버 정보는 리턴 값, 오류 아이디 등을 포함하고, 프로세스 정보는 프로세스 아이디 및 이름을 포함한다.

[0038] 다음, S150 단계에서 수집된 테스트 정보가 분석된다(S160). S160 단계의 분석 결과에 따라, 어플리케이션이 예외처리를 제대로 하는지 확인할 수 있다.

[0039] 본 발명에 따른 결함 모델은 어플리케이션과 디바이스 사이의 연결, 디바이스 오픈, 연결해제, 디바이스 닫기, 데이터 입출력과 관련하여 널 데이터(NULL)와 데이터 오류와 처리지연, 또는 디바이스 전원과 관련된 결함 등을 포함한다.

[0040] 연결과 관련된 결함은 디바이스 연결 초기화 실패, 디바이스 오픈 실패로 분류될 수 있다.

[0041] 연결해제와 관련된 결함은 디바이스 종료화 실패, 디바이스 연결 해제 실패로 분류될 수 있다.

[0042] 디바이스 닫기와 관련된 결함은 디바이스 닫기 실패로 정의될 수 있다.

[0043] 널(NULL) 데이터와 관련된 결함은 디바이스로부터의 데이터 읽기 실패, 디바이스로의 데이터 쓰기 실패, 디바이스로의 데이터 탐색 실패, 디바이스로부터의 IOCTL 데이터 전달 실패로 분류될 수 있다.

[0044] 데이터 오류와 관련된 결함은 디바이스로부터의 잘못된 데이터 읽기, 디바이스로의 잘못된 데이터 읽기, 디바이스로의 잘못된 데이터 탐색, 디바이스로의 잘못된 값을 IOCTL 전송, 디바이스로부터의 잘못된 값을 IOCTL 전송으로 분류될 수 있다.

[0045] 디바이스 전원과 관련된 결함은 디바이스 파워 업 실패, 디바이스 파워 다운 실패로 분류될 수 있다.

[0046] 본 발명은 프로세서가 읽을 수 있는 기록매체에 프로세서가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 프로세서가 읽을 수 있는 기록매체는 프로세서에 의해 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 프로세서가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광 데이터 저장장치 등이 있다. 또한 프로세서가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 프로세서가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

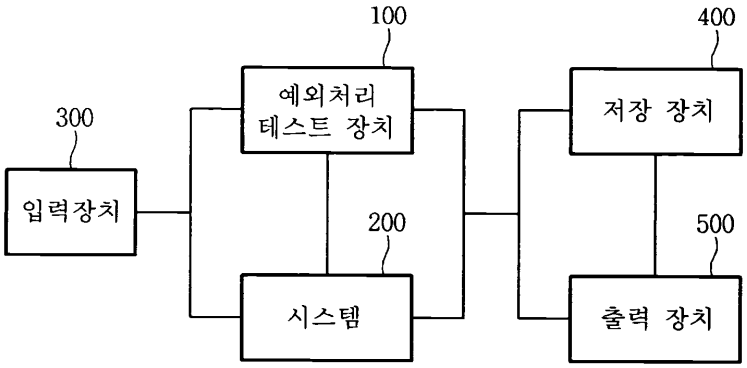
[0047] 본 발명은 상술한 일실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형실시가 가능하며, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어서는 안될 것이다.

[0048] **산업상 이용가능성**

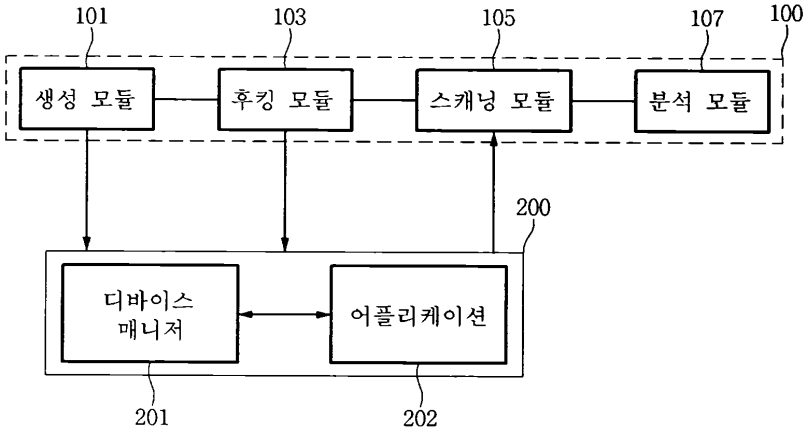
[0049] 본 발명에 따르면, 시스템이 실제로 운용되는 동안에도 디바이스 매니저를 이용한 예외처리 테스트가 수행될 수 있다.

도면

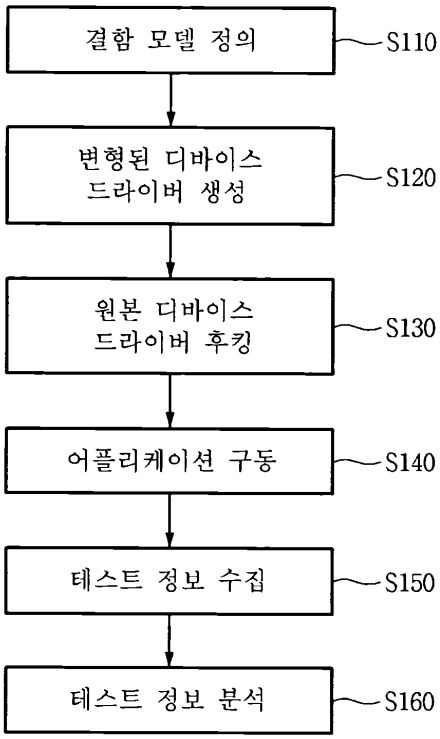
도면1



도면2


101
103
105
107
100
생성 모듈
후킹 모듈
스캐닝 모듈
분석 모듈
200
디바이스
매니저
어플리케이션
201
202

도면3


결함 모델 정의
S110
변형된 디바이스
드라이버 생성
S120
원본 디바이스
드라이버 후킹
S130
어플리케이션 구동
S140
테스트 정보 수집
S150
테스트 정보 분석
S160