

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2013-0042502 (43) 공개일자 2013년04월26일
(51) 국제특허분류(Int. Cl.) G06F 11/22 (2006.01)	(71) 출원인 현대자동차주식회사 서울특별시 서초구 현릉로 12 (양재동) 이화여자대학교 산학협력단 서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교) 기아자동차주식회사 서울특별시 서초구 현릉로 12 (양재동)	
(21) 출원번호 10-2012-7034163	(72) 발명자 최병주 서울특별시 강남구 압구정동 현대아파트 74-1305 서주영 서울특별시 송파구 문정동 올림픽훼밀리타운 105-304 (뒷면에 계속)	
(22) 출원일자(국제) 2011년03월15일 심사청구일자 2012년12월27일	(74) 대리인 특허법인태평양	
(85) 번역문제출일자 2012년12월27일		
(86) 국제출원번호 PCT/KR2011/001803		
(87) 국제공개번호 WO 2012/002635 국제공개일자 2012년01월05일		
(30) 우선권주장 1020100061455 2010년06월28일 대한민국(KR) PCT/KR2010/006068 2010년09월07일 세계지적재산권기구(WIPO)(WO)		

전체 청구항 수 : 총 20 항

(54) 발명의 명칭 시스템 테스트 장치

(57) 요약

본 발명은 시스템 테스트 장치로서, 본 발명은 시스템 테스트 장치로서, 프로세스 제어블록에 테스트 에이전트를 삽입하는 삽입 모듈, 테스트 대상과 관련된 이벤트가 발생하는 경우, 상기 테스트 에이전트를 이용하여 테스트 대상을 테스트 코드로 후킹하는 후킹 모듈, 상기 테스트 대상이 후킹된 경우, 상기 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보를 수집하는 스캐닝 모듈, 및 상기 스캐닝 모듈에 의해 수집된 테스트 정보를 저장하는 로깅 모듈을 포함한다.

(72) 발명자

양승완

경기도 군포시 광정동 삼성장미아파트 1137-2202

임진용

경기도 수원시 권선구 금곡동 삼익 1차 아파트 10
2동 406호

김영수

경기도 용인시 기흥구 신갈동 697번지 녹원마을 새
천년그린빌 509동 503호

권해영

경기도 용인시 기흥구 보정동 현대아이파크 1차 아
파트 202-601

오정석

경기도 광명시 광명1동 70-13, 2층

장승연

경기도 고양시 덕양구 화정2동 별빛마을 건영아파
트 1004동 1101호

특허청구의 범위

청구항 1

복수의 프로세스 및 상기 복수의 프로세스에 대한 런-타임 실행정보를 포함하는 프로세스 제어블록을 포함하는 시스템에 대한 시스템 테스트 장치에 있어서,

상기 프로세스 제어블록에 테스트 에이전트를 삽입하는 삽입 모듈;

테스트 대상과 관련된 이벤트가 발생하는 경우, 상기 테스트 에이전트를 이용하여 테스트 대상을 테스트 코드로 후킹하는 후킹 모듈;

상기 테스트 대상이 후킹된 경우, 상기 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보를 수집하는 스캐닝 모듈; 및

상기 스캐닝 모듈에 의해 수집된 테스트 정보를 저장하는 로깅 모듈을 포함하는 시스템 테스트 장치.

청구항 2

제1항에 있어서,

상기 프로세스 제어블록은 상기 복수의 프로세스에 대한 자원사용 현황, 프로세스 상태, 우선순위 및 함수테이블 중 적어도 하나를 포함하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 3

제2항에 있어서,

상기 테스트 코드는 결함 유형에 따라 정의되는 것을 특징으로 하는 시스템 테스트 장치.

청구항 4

제3항에 있어서,

상기 후킹 모듈은 상기 결함 유형에 따른 테스트 위치에 따라 메모리 영역을 식별하여, 식별된 메모리 영역을 테스트 코드로 후킹하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 5

제3항에 있어서,

상기 테스트 정보는 프로그램 실행 정보, 프로세스 컨텍스트 정보 및 시스템 상태 정보 중 적어도 하나인 것을 특징으로 하는 시스템 테스트 장치.

청구항 6

제2항에 있어서,

상기 스캐닝 모듈은 상기 프로세스 제어블록의 성능 정보를 측정하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 7

제6항에 있어서,

상기 프로세스 제어블록의 성능 정보는 프로세서 사용량, 메모리 사용량, 페이지 폴트(Page Fault), 프로세스 컨텍스트 정보, 프로세스 콜스택, 프로세스 힙 사용량, 프로세스 상태, 프로세스 우선순위 중 적어도 하나인 것을 특징인 시스템 테스트 장치.

청구항 8

제6항에 있어서,

미리 설정된 시간 간격마다 상기 프로세스 제어블록의 성능 정보를 측정하는 것을 특징으로 하는 시스템 테스트

장치.

청구항 9

제6항에 있어서,

상기 프로세스 제어 블록의 프로세스 및 스레드(Thread)에 대한 성능 정보를 측정하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 10

제9항에 있어서,

상기 프로세스 제어 블록의 프로세스에 대한 성능 요소는

아이디(ID), 프로세스의 상태, 프로세스의 우선 순위, 힙(Heap) 사용량, 프로세스의 동작 시간, 사용 시간, 커널 사용 시간 중 하나 이상을 포함하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 11

제9항에 있어서,

상기 프로세스 제어 블록의 스레드에 대한 성능 요소는

아이디, 운영 상태(Run State), 기본 우선 순위, 현재 우선 순위, 사용 시간, 커널 사용 시간, 콜 스택(Call Stack) 중 하나 이상을 포함하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 12

제2항에 있어서,

상기 테스트 대상과 관련된 이벤트는 상기 복수의 프로세스 사이의 메시지 전송, 공유 데이터 전송, 공유 메모리 사용, 메시지큐 사용, 동기화 객체 사용 중 적어도 하나인 것을 특징으로 하는 시스템 테스트 장치.

청구항 13

제12항에 있어서,

상기 테스트 코드는 상기 복수의 프로세스 사이의 메시지 전송, 메시지 큐, 공유 메모리 및 동기화 객체 중 적어도 하나의 결합 유형을 확인하기 위한 코드인 것을 특징으로 하는 시스템 테스트 장치.

청구항 14

제12항에 있어서,

상기 테스트 정보는 상기 복수의 프로세스의 콜-스택 정보, 상기 복수의 프로세스 사이에 송수신되는 통신 유형, 통신 아이디, 통신 데이터 및 상기 복수의 프로세스 사이의 통신에서 발생한 지연시간 중 적어도 하나인 것을 특징으로 하는 시스템 테스트 장치.

청구항 15

제12항에 있어서,

상기 테스트 정보는 시스템 동적 상태 정보, 프로세스 상태 정보, 및 상기 복수의 프로세스 사이의 통신 정보를 포함하는 것을 특징으로 하는 시스템 테스트 장치.

청구항 16

복수의 프로세스 및 상기 복수의 프로세스에 대한 런-타임 실행정보를 포함하는 프로세스 제어블록을 포함하는 시스템에 대한 시스템 테스트 시스템에 있어서,

상기 시스템을 구동시키기 위한 신호를 입력받는 입력 장치;

상기 입력 장치를 통해 입력된 신호에 의해 상기 시스템이 동작하는 동안, 상기 프로세스 제어블록에 테스트 에

이전트를 삽입하고, 상기 테스트 에이전트를 이용하여 테스트 대상을 테스트 코드로 후킹하고, 상기 복수의 프로세스 중 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보를 수집하는 시스템 테스트 장치; 및

상기 시스템 테스트 장치에 의해 수집된 테스트 정보를 분석하는 분석 장치를 포함하는 시스템 테스트 시스템.

청구항 17

제16항에 있어서,

상기 프로세스 제어블록은 상기 복수의 프로세스에 대한 자원사용 현황, 프로세스 상태, 우선순위 및 함수테이블 중 적어도 하나를 포함하는 것을 특징으로 하는 시스템 테스트 시스템.

청구항 18

제16항에 있어서,

상기 테스트 코드가 후킹되는 테스트 위치는 결함 유형에 따라 변경될 수 있는 것을 특징으로 하는 시스템 테스트 시스템.

청구항 19

제16항에 있어서,

상기 테스트 정보는 프로그램 실행 정보, 프로세스 컨텍스트 정보, 시스템상태정보, 프로세서 사용량, 메모리 사용량, 페이지 폴트(Page Fault), 상기 복수의 프로세스의 콜-스택 정보, 상기 복수의 프로세스 사이에 송수신되는 통신 유형, 통신 아이디, 통신 데이터 및 상기 복수의 프로세스 사이의 통신에서 발생한 지연시간 중 적어도 하나인 것을 특징으로 하는 시스템 테스트 시스템.

청구항 20

제16항에 있어서,

상기 분석 장치에 의해 분석된 테스트 결과를 이미지 또는 음성으로 출력하는 출력 장치를 더 포함하는 시스템 테스트 시스템.

명세서

기술 분야

[0001] 본 발명은 시스템 테스트 장치에 관한 것으로, 보다 상세하게는 시스템의 실행 정보를 포함하는 프로세스 제어블록을 이용하여 시스템에 대한 테스트 정보를 실시간으로 수집하는 시스템 테스트 장치에 관한 것이다.

배경 기술

[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 중요하므로, 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 시스템에 대한 메모리 오류 테스트, 성능 테스트 및 시스템 내부에 포함된 프

로세스 사이의 통신 테스트를 수행함으로써, 실제 운영되는 시스템에 대한 테스트가 가능하고, 시스템의 런-타임 상황에서도 동적 테스트를 수행할 수 있고, 시스템이 동작하는 동안 발생하는 결함을 확인할 수 있는 시스템 테스트 장치를 제공하는데 그 목적이 있다.

과제의 해결 수단

[0005] 본 발명은 시스템 테스트 장치로서, 프로세스 제어블록에 테스트 에이전트를 삽입하는 삽입 모듈, 테스트 대상과 관련된 이벤트가 발생하는 경우, 상기 테스트 에이전트를 이용하여 테스트 대상을 테스트 코드로 후킹하는 후킹 모듈, 상기 테스트 코드로 후킹된 경우, 상기 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보를 수집하는 스캐닝 모듈, 및 상기 스캐닝 모듈에 의해 수집된 테스트 정보를 저장하는 로깅 모듈을 포함한다.

[0006] 또한, 본 발명은 시스템 테스트 시스템으로서, 시스템을 구동시키기 위한 신호를 입력받는 입력 장치, 상기 입력 장치를 통해 입력된 신호에 의해 상기 시스템이 동작하는 동안, 상기 프로세스 제어블록에 테스트 에이전트를 삽입하고, 상기 테스트 에이전트를 이용하여 테스트 대상을 테스트 코드로 후킹하고, 상기 복수의 프로세스 중 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보를 수집하는 시스템 테스트 장치 및 상기 시스템 테스트 장치에 의해 수집된 테스트 정보를 분석하는 분석 장치를 포함한다.

발명의 효과

[0007] 본 발명에 따르면, 소스 코드가 아닌 실제 운영되는 코드를 대상으로 하므로, 시스템이 실제로 운용되는 동안에도 시스템에 대한 테스트를 수행할 수 있다. 따라서, 시스템이 실제 동작하는 동안 메모리 오류, 시스템 성능, 프로세스 사이의 통신을 테스트 할 수 있다.

[0008] 또한, 각각의 프로세스에 대한 개별적인 코드 삽입이 아닌 프로세스 제어블록에 포함된 정보를 활용한 후킹 방법을 이용하므로, 시스템 오버헤드를 최소화할 수 있고, 동작하는 시스템을 방해하지 않고 시스템에 대한 테스트를 수행할 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 시스템 테스트 장치가 이용되는 전체 시스템을 나타내는 도면이다.

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

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

도 4는 본 발명의 일실시예에 따른 메모리 테스트 방법, 성능 테스트 방법 및 프로세스 사이의 통신 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 후킹 기법을 이용하여, 시스템 내에서 동작하는 모든 프로세스에 대한 테스트를 수행할 수 있다.

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

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

[0014] 시스템 테스트 장치(100)는 시스템(200)에 대한 메모리 결함, 시스템(200)의 성능 또는 시스템(200)에 포함된 둘 이상의 프로세스 사이의 통신을 테스트하기 위한 장치이다. 시스템 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] 시스템(200)은 시스템 테스트 장치(100)의 테스트 대상이 되는 시스템으로서, 다양한 소프트웨어 및 하드웨어가 통합된 임베디드 시스템이다. 시스템 테스트 장치(100)의 테스트 대상이 되는 시스템은 복수의 프로세스를 포함하는 하나 이상의 시스템을 의미한다. 따라서, 복수의 시스템의 메모리 결함 또는 시스템 성능도 테스트 대상이

될 수 있으며, 하나의 시스템에 포함되는 복수의 프로세스 사이의 통신뿐 아니라, 서로 다른 시스템에 포함된 프로세스들 사이의 통신도 시스템 테스트 장치(100)의 테스트 대상이 될 수 있다.

- [0016] 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), VDC (Vehicle CDC), DIS (Driver Information System), RSE (Rear Seat Entertainment), HEV (Hybrid Electric Vehicle) display, AHU (Air Handling Unit) display 등을 포함한다.
- [0017] 입력 장치(300)는 시스템 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)이 구동된다. 또한, 입력 장치(300)를 통해 입력되는 신호에 따라 시스템 테스트 장치(100)가 시스템 테스트를 위한 동작을 실행한다.
- [0018] 분석 장치(400)는 시스템 테스트 장치(100)에 의해 수집된 테스트 정보를 분석하여 시스템(200)의 테스트 결과를 확인한다. 상기 시스템 테스트 장치(100)는 분석 장치(400)와 하나의 장치로 구현되어, 테스트 정보를 수집함과 동시에 수집된 테스트 정보를 분석하여 테스트 결과를 확인할 수도 있다.
- [0019] 출력 장치(500)는 시스템 테스트 장치(100)에 의해 수집된 테스트 정보 또는 분석 장치(400)에 의해 분석된 테스트 결과를 이미지 또는 음성으로 출력하는 장치이다. 출력 장치(500)를 통해 사용자는 시스템 테스트 결과를 확인할 수 있다.
- [0020] 도 2는 본 발명의 일실시예에 따른 시스템 테스트 장치를 나타내는 구성도이다.
- [0021] 도 2에 도시된 바와 같이, 시스템 테스트 장치(100)는 삽입 모듈(101), 후킹 모듈(103), 스캐닝 모듈(105) 및 로깅 모듈(107)을 포함한다. 도 2에서 시스템 테스트 장치(100)와 연결되어 테스트 대상이 되는 시스템(200)은 둘 이상의 프로세스를 포함하는 하나의 시스템으로 가정한다.
- [0022] 시스템(200)은 프로세스 제어블록(201, Process Control Block)을 포함한다. 프로세스 제어블록(201)은 시스템(200)에 포함된 모든 프로세스에 대한 런-타임 실행정보를 갖는다. 여기서, 런-타임 실행정보란, 시스템(200)이 동작하는 동안 모든 프로세스 사이의 실행정보를 의미한다. 따라서, 프로세스 제어블록(201)은 시스템(200)에 포함된 모든 프로세스에 대한 자원사용 현황, 프로세스 상태, 우선순위, 함수테이블 등을 포함할 수 있다. 보다 상세하게는 프로세스 ID, 프로세스 이름, 현행 프로세스, 현행 쓰레드, 시스템에서 실행되고 있는 프로세스 리스트, 모듈 리스트, 시스템 페이지 사이즈, 사용 가능한 물리메모리 사이즈, CPU 사용량, 사용중인 자원 리스트, 요청중인 I/O디바이스 상태, 오픈된 파일 정보, 레지스터 정보 등을 포함할 수 있다. 따라서, 모든 시스템 내에서 상기 언급된 정보를 포함하는 데이터구조 또는 장치는 본 발명에서 의미하는 프로세스 제어블록에 해당된다.
- [0023] 삽입 모듈(101)은 프로세스 제어블록(201)에 테스트 에이전트를 삽입한다. 테스트 에이전트는 프로세스 제어블록(201)에 포함된 정보를 기초로 테스트 대상을 테스트 코드로 후킹하고, 테스트 정보를 수집할 수 있다. 테스트 코드가 후킹되는 테스트 위치는 결함 유형에 따라 정의될 수 있다. 본 발명에서, 테스트 코드로 후킹되는 테스트 대상은 테스트 대상이 되는 메모리 영역, 시스템의 성능 정보를 포함하는 코드, 또는 통신 관련 이벤트를 발생시키는 코드 등을 될 수 있다. 본 발명에서 테스트 에이전트가 프로세스 제어블록(201)에 삽입되기 전에, 먼저 시스템(200)은 정상적으로 동작하고 있으며, 시스템(200)은 시스템 테스트 장치(100)의 동작에 의해 영향을 받지 않고 정상적으로 동작한다.
- [0024] 또한, 삽입 모듈(101)은 시스템(200)에서 동작하고 있는 프로세스 리스트를 확인하여, 확인된 프로세스 리스트를 기초로 테스트 에이전트를 삽입할 수 있다.
- [0025] 후킹 모듈(103)은 상기 삽입 모듈(101)에 의해 삽입된 테스트 에이전트를 이용하여 테스트 대상을 테스트 코드로 후킹한다. 상기 테스트 코드는 테스트하고자 하는 대상 및 결함 유형에 따라 변경된다. 즉, 메모리 결함 테스트, 성능 테스트 및 통신 테스트에 따라 테스트 코드가 각각 달라지며, 통신 테스트에서도 테스트하고자 하는 결함 유형에 따라 테스트 코드가 달라질 수 있다. 또한, 후킹 모듈(103)은 테스트 대상이 테스트 코드로 후킹되는 테스트 위치를 변경할 수 있다.

- [0026] 테스트 코드가 후킹된다는 것은, 동작하고 있는 시스템(200) 내에서 테스트 대상과 관련 이벤트가 발생하는 경우, 삽입된 테스트 에이전트가 원본 코드의 동작을 포함한 테스트 코드가 실행되도록 OS의 프로세스 제어블록을 후킹한다는 것을 의미한다. 프로세스 제어블록에 포함되는 정보로부터 후킹 모듈(103)은 테스트 대상과 관련된 이벤트가 발생하는지 확인할 수 있다.
- [0027] 스캐닝 모듈(105)은 테스트 코드가 후킹된 경우, 상기 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보를 수집한다. 여기서, 테스트 정보는 기발생한 이벤트에 대한 정보 뿐 아니라, 테스트 코드 후킹에 따라 새롭게 발생하는 이벤트까지 포함할 수 있다. 수집되는 테스트 정보는 테스트 단계, 테스트 목적, 테스트 하고자 하는 결함 유형에 따라 달라진다.
- [0028] 따라서, 스캐닝 모듈(105)은 시스템 테스트 결과 분석에 필요한 시스템 동적 상태 정보, 예를 들어, 함수 파라미터 및 반환 값, 성능 측정값, 공유 자원 사용 리스트 등을 수집한다.
- [0029] 로깅 모듈(107)은 상기 스캐닝 모듈(105)에 의해 수집된 테스트 정보를 저장 공간에 저장한다. 저장 공간은 로깅 모듈(107)에 구비될 수도 있고, 로깅 모듈(107)과 별개 장치로 구현될 수 있다. 도 1의 분석 장치(400)는 스캐닝 모듈(105)에 의해 수집된 테스트 정보 또는 로깅 모듈(107)에 의해 저장된 테스트 정보를 분석할 수 있다.
- [0030] 도 3은 본 발명의 일실시예에 따른 시스템 테스트 방법을 나타내는 흐름도이다.
- [0031] 도 3에 도시된 바와 같이, 테스트 대상이 되는 시스템(200)의 프로세스 제어블록(201)에 테스트 에이전트가 삽입된다(S110). 시스템(200)은 테스트 에이전트가 삽입되기 전부터 정상적으로 동작하고 있다.
- [0032] 다음, 테스트 대상과 관련된 이벤트가 발생하는 경우, 삽입된 테스트 에이전트를 통해 테스트 대상이 테스트 코드로 후킹된다(S120). 테스트 대상이 테스트 코드로 후킹됨으로써, 상기 이벤트가 발생한 프로세스 대신 테스트 코드가 실행될 수 있다.
- [0033] 다음, 테스트 대상이 테스트 코드로 후킹된 경우, 상기 테스트 대상과 관련된 이벤트가 발생한 프로세스에 대한 테스트 정보가 수집된다(S130). 다음, 수집된 테스트 정보는 저장된다(S140). 여기서, 테스트 정보는 테스트 대상이 테스트 코드로 후킹되고 있는 동안에만 수집될 수 있고, 테스트 대상이 테스트 코드로 후킹된 후, 시스템이 동작하는 동안 모두 수집될 수도 있다.
- [0034] 도 4는 본 발명의 일실시예에 따른 메모리 결함 테스트 방법, 성능 테스트 방법 및 프로세스 사이의 통신 테스트 방법을 나타내는 흐름도이다.
- [0035] 도 4a는 메모리 결함 테스트를 수행하기 위한 시스템 테스트 방법을 나타낸다. 먼저, 결함 모델이 정의된다(S150). 여기서, 결함 모델이란, 결함 유형에 따라 테스트되어야 하는 메모리 위치 및 결함 발견/복구를 위한 테스트 정보를 의미한다. 정의된 결함 모델에 따라, 시스템(200)이 정상적으로 동작하는 동안 테스트 정보를 수집할 수 있는 테스트 코드가 생성될 수 있다.
- [0036] 다음, 시스템(200)의 프로세스 제어블록(201)에 테스트 에이전트가 삽입된다(S151). 테스트 에이전트는 시스템(200)에서 동작하고 있는 프로세스 리스트 중 적어도 하나에 삽입될 수 있다.
- [0037] 다음, 후킹 모듈(103)은 프로세스 제어블록(201)에 포함된 정보 및 결함 모델을 기초로 후킹할 메모리 영역을 식별하고, 식별된 메모리 영역에 테스트 코드를 후킹한다(S152). 메모리 영역을 식별한 결과, 테스트 위치가 SYSTEM API(Application Process Interface) 함수의 경우, 후킹 모듈(103)은 프로세스 제어블록(201)의 APISet을 후킹 영역으로 식별한다. 테스트 위치가 DLL(Dynamic Linking Library) EXPORT 함수의 경우, 후킹 모듈(103)은 프로세스 제어블록(201)의 IAT (Import Address Table)을 후킹 영역으로 식별한다. 그 외의 경우, 후킹 모듈(103)은 프로세스 제어블록(201)의 코드 공간을 후킹 영역으로 식별한다.
- [0038] 도 4a에서, 테스트 코드가 후킹된다는 것은, 동작하고 있는 시스템(200) 내에서 테스트 대상인 메모리 위치에, 메모리에 대한 쓰기, 읽기, 접근 등의 이벤트가 발생하는 경우, 삽입된 테스트 에이전트가 원본 코드의 동작을 포함한 테스트 코드가 실행되도록 OS의 프로세스 제어블록을 후킹한다는 것을 의미한다.
- [0039] 다음, 테스트 대상인 메모리 상태에 대한 정보가 수집된다(S153). 예를 들어, 결함 원인 식별에 필요한 프로그램 실행 정보가 수집되고, 결함 위치 식별에 필요한 프로세스 컨텍스트 정보가 수집되며, 메모리 성능 분석에 필요한 시스템 상태 정보가 수집될 수 있다.
- [0040] 다음, S153 단계에서 수집된 테스트 정보가 저장된다(S154).

- [0041] 도 4b는 성능 테스트를 수행하기 위한 시스템 테스트 방법을 나타낸다. 본 발명에 따른 시스템 테스트 방법은 성능 병목과 그 원인을 분석하기 위해 프로세스 제어블록(201)의 데이터 중 페이지 폴트율, 프로세서 사용율과 같은 시스템 실행 정보들을 후킹하여 시스템 성능을 분석할 수 있다.
- [0042] 여기서, 성능은 시스템이나 컴포넌트들이 주어진 시스템의 제약 조건 하에서 기능을 수행하는 정도를 의미한다. 성능 테스트는 특정 성능 요구사항을 시스템이 만족하는 지에 관한 평가이다. 이러한 성능 테스트를 통해 성능 병목과 그 원인을 분석하여 이를 해결함으로써 시스템의 성능을 개선할 수 있다.
- [0043] 시스템의 성능 병목은 메모리, I/O 디바이스, 네트워크와 같은 제한된 자원들에 대한 경쟁으로 시스템의 성능이 저하되는 현상이다. 시스템 성능 병목의 원인은 자원 부족, 공유자원 경쟁, 자원 독점, 자원의 잘못된 구성 또는 자원의 잘못된 동작과 같이 다양하다.
- [0044] 먼저, 결합 모델이 정의된다(S160). 여기서, 결합 모델이란, 결합 유형에 따라 테스트되어야 하는 성능 병목 위치 및 성능 병목을 판정할 시스템 상태를 나타내는 테스트 정보를 의미한다. 정의된 결합 모델에 따라, 시스템 (200)이 정상적으로 동작하는 동안 테스트 정보를 수집할 수 있는 테스트 코드가 생성될 수 있다.
- [0045] 삽입 모듈(101)은 프로세스 제어블록(201)에 테스트 에이전트를 삽입한다(S161). 삽입 모듈(101)은 시스템(200)에서 동작하고 있는 프로세스 리스트를 확인하여, 확인된 프로세스 리스트를 기초로 테스트 에이전트를 삽입할 수 있다. 즉, 현행 프로세스(Current Process)에 관한 프로세스 제어블록(201)의 base address는 특정 메모리 공간에 관리되거나, 경우에 따라 미리 고정된 주소로 관리되기 때문에 현행 프로세스의 프로세스 제어블록(201)의 정보를 알 수 있다.
- [0046] 다음, 후킹 모듈(103)은 테스트 대상에 테스트 코드를 후킹한다(S162).
- [0047] 테스트 대상이 테스트 코드로 후킹된 경우, 스캐닝 모듈(105)은 시스템의 성능 정보를 수집한다(S163). 예를 들어, timer-interrupt를 사용하며 지정된 시간 간격 (e.g. 1sec, 100msec) 마다 성능 데이터를 측정할 수 있다. 또한, 시스템 성능을 시스템 단위뿐만 아니라 프로세스와 스레드의 단위로까지 분석하기 위해, 프로세스 제어블록(201)의 프로세스와 스레드 리스트들을 순환하면서 성능 정보를 측정할 수 있다.
- [0048] 시스템 성능 병목이 발생하는 위치를 자세히 추적하기 위해, 스캐닝 모듈(105)은 스레드 별로 콜-스택(Call-stack) 정보를 저장할 수 있으며, 각 모니터링 단위 별 측정되는 성능 정보는 다음과 같다.
- [0049] 프로세스 제어블록(201)의 성능 정보는 프로세서 사용량, 메모리 사용량, 페이지 폴트(Page Fault), 프로세스 컨텍스트 정보, 프로세스 콜스택, 프로세스 힙 사용량, 프로세스 상태, 프로세스 우선순위 중 하나 이상을 포함할 수 있다. 프로세스 제어블록(201)의 프로세스에 대한 성능 정보는 아이디(ID), 프로세스의 상태, 프로세스의 우선 순위, 힙(Heap) 사용량, 프로세스의 동작 시간, 사용 시간, 커널 사용 시간 중 하나 이상을 포함할 수 있다. 또한, 프로세스 제어블록(201)의 스레드에 대한 성능 정보는 아이디, 운영 상태(Run State), 기본 우선 순위, 현재 우선 순위, 사용 시간, 커널 사용 시간, 콜스택 정보 중 하나 이상을 포함할 수 있다.
- [0050] 다음, 로깅 모듈(107)은 수집된 테스트 정보를 저장 공간에 저장한다(S164).
- [0051] 성능 테스트를 수행하기 위한 시스템 테스트 방법은 페이지폴트를 기준으로 메모리 병목을 판단할 수 있다. 예를 들어, 페이지 폴트가 높으면 메모리 병목일 수 있다. 페이지폴트란 프로그램이 자신의 주소 공간에는 존재하지만 시스템의 메모리에는 현재 없는 데이터나 코드에 접근 시도하였을 경우 발생하는 현상이다. 페이지폴트가 발생하면 운영체제는 그 데이터를 메모리로 가져와서 마치 페이지폴트가 전혀 발생하지 않은 것처럼 프로그램이 계속적으로 작동하도록 한다. 운영체제의 페이지폴트에 대한 예외처리 때문에 어플리케이션의 처리시간은 지연되며, 전체 시스템 성능에 영향을 준다.
- [0052] 또한, 본 발명에 따른 시스템 테스트 방법은 메모리 사용량을 통해 성능 병목을 식별할 수 있다. 시스템의 메모리는 물리 메모리 사용량, 가상 메모리 사용량이 있고, 프로세스 별로 힙 메모리 사용량으로 구분할 수 있다. 본 발명에 따른 시스템 테스트 방법은 이러한 메모리 사용량들의 합을 기준으로 성능 병목을 판단한다.
- [0053] 또한 본 발명에 따른 시스템 테스트 방법은 프로세서 사용량 (혹은 CPU 사용량)을 통해 성능 병목을 식별할 수 있다. 예를 들어, 본 발명에 따른 시스템 테스트 방법은 프로세서 사용량이 높게 유지되면서 여유 메모리가 있는 경우는 CPU에 병목이 있다고 판단할 수 있다. 반면 프로세서 사용량이 높고 메모리도 고갈되었다면 성능 문제는 CPU보다 메모리 병목으로 판단할 수 있다.
- [0054] 또한 본 발명에 따른 시스템 테스트 방법은 프로세스 사용량 (Processor Usage)에 따라 성능 병목을 식별할 수

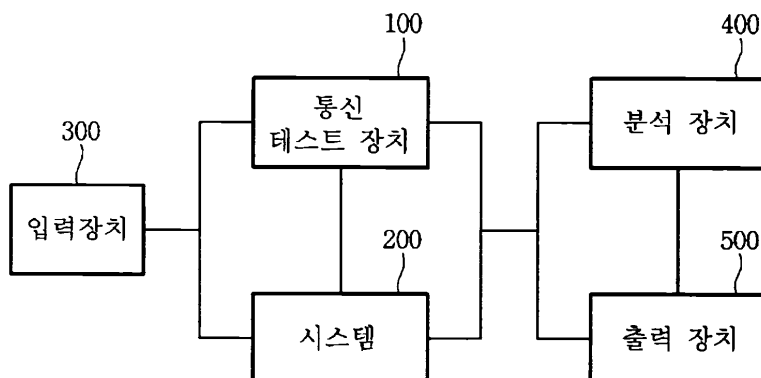
있다. 프로세스 사용량은 시스템의 실행 시간으로 전체 CPU 사용량에서 대기 시간(Idle Time)을 제외한 시간을 의미한다.

- [0055] 또한 본 발명에 따른 시스템 테스트 방법은 사용 시간 (User Time)에 따라 성능 병목을 판단할 수 있다. 사용 시간은 실행이 사용자 공간에서 머무른 시간을 의미하며, 곧 어플리케이션의 실행 시간을 의미한다.
- [0056] 또한 본 발명에 따른 시스템 테스트 방법은 커널 사용 시간 (Kernel Time)에 따라 성능 병목을 판단할 수 있다. 커널 사용 시간은 실행이 커널 공간에서 머무른 시간으로, 커널의 서비스 처리 시간을 의미한다.
- [0057] 도 4c는 복수의 프로세스 사이의 통신에 대한 테스트를 수행하기 위한 시스템 테스트 방법을 나타낸다. 도 4c에 따른 시스템 방법을 수행하는 시스템 테스트 장치(100)는 시스템(200)에 포함된 모든 프로세스 사이의 자원 공유 및 의사 전달이 정확히 이루어지는지 판단할 수 있는 테스트 정보를 수집할 수 있다.
- [0058] 먼저, 결함 모델이 정의된다(S170). 여기서, 결함 모델이란, 결함 유형에 따라 테스트되어야 하는 프로세스 사이의 통신 위치 및 결함 발견/복구를 위한 테스트 정보를 의미한다. 정의된 결함 모델에 따라, 시스템(200)이 정상적으로 동작하는 동안 테스트 정보를 수집할 수 있는 테스트 코드가 생성될 수 있다.
- [0059] 테스트 대상이 되는 시스템(200)의 프로세스 제어블록(201)에 테스트 에이전트가 삽입된다(S171). 삽입 모듈(101)은 상기 프로세스 리스트 중 둘 이상의 프로세스를 테스트 대상으로 선정하여, 선정된 프로세스 사이의 통신을 테스트하기 위한 테스트 에이전트를 삽입할 수도 있다.
- [0060] 다음, 복수의 프로세스 사이에 통신 관련 이벤트가 발생하는 경우, 삽입된 테스트 에이전트를 이용하여 테스트 대상이 테스트 코드로 후킹된다(S172).
- [0061] 테스트 대상이 테스트 코드로 후킹된다는 것은, 동작하고 있는 시스템(200) 내에서 복수의 프로세스 사이에 통신 관련 이벤트가 발생하는 경우, 삽입된 테스트 에이전트가 원본 코드의 동작을 포함한 테스트 코드가 실행되도록 OS의 프로세스 제어블록을 후킹한다는 것을 의미한다. 프로세스 제어블록에 포함되는 정보로부터 후킹 모듈(103)은 어떤 프로세스 사이에 어떤 유형의 통신 관련 이벤트가 발생하는지를 확인할 수 있다. 여기서, 통신 관련 이벤트란, 복수의 프로세스 사이에 메시지 전송, 공유 데이터 전송, 공유 메모리 사용, 메시지큐 사용, 동기화 객체 사용 등을 의미한다.
- [0062] 테스트 코드는 상기 복수의 프로세스 사이의 메시지 전송, 메시지 큐, 공유 메모리 또는 동기화 객체에 관련된 결함 유형을 확인하기 위한 코드이다.
- [0063] 예를 들어, 테스트 코드는 SendMessage(), PostMessage()와 같은 제1함수 및 테스트 하고자 하는 결함유형에 따라 필요한 제2함수를 포함하며, 시스템에 포함된 프로세스들은 각각 제1함수를 포함한다. 따라서, 제1 프로세스 및 제2 프로세스 사이에 통신 관련 이벤트가 발생하는 경우, 후킹 모듈(103)은 삽입된 통신 에이전트를 이용하여 제1 프로세스 및 제2 프로세스에서 제1함수가 실행되는 단계에서, 제1 프로세스 및 제2 프로세스의 제1함수 대신 테스트 코드를 실행시킴으로써, 테스트 코드를 후킹한다. 제1 프로세스 및 제2 프로세스에서 제1함수가 실행되는 단계는 프로세스 제어블록에 포함된 정보로부터 확인된다.
- [0064] 한편, 테스트 대상이 복수의 프로세스 사이의 통신이므로 테스트하고자 하는 결함 유형에 따라 송신 측 또는 수신 측에 해당하는 일방의 프로세스에만 테스트 코드를 후킹할 수도 있다.
- [0065] 다음, 스캐닝 모듈(105)은 테스트 대상이 후킹된 경우, 상기 통신 관련 이벤트가 발생한 복수의 프로세스 사이의 통신에 대한 테스트 정보를 수집한다(S172). 여기서, 복수의 프로세스 사이의 통신이란, 기발생한 통신 관련 이벤트 뿐 아니라, 테스트 코드 후킹에 따라 발생하는 통신 관련 이벤트까지 포함한다. 수집되는 테스트 정보는 테스트 단계, 테스트 목적, 테스트하고자 하는 결함 유형에 따라 달라진다.
- [0066] 따라서, 스캐닝 모듈(105)은 복수의 프로세스 사이의 통신에 대해 결함 판정에 필요한 시스템 동적 상태 정보, 예를 들어, 함수 파라미터 및 반환 값, 성능 측정값, 공유 자원 사용 리스트 등을 수집한다.
- [0067] 테스트 정보는 상기 복수의 프로세스의 콜-스택 정보, 상기 복수의 프로세스 사이에 송수신되는 통신 유형, 통신 아이디, 통신 데이터, 상기 복수의 프로세스 사이의 통신에서 발생한 지연시간 등을 포함할 수 있다.
- [0068] 다음, 로깅 모듈(107)은 상기 스캐닝 모듈(105)에 의해 수집된 테스트 정보를 저장 공간에 저장한다(S173). 저장 공간은 로깅 모듈(107)에 구비될 수도 있고, 로깅 모듈(107)과 별개 장치로 구현될 수 있다. 도 1의 분석 장치(400)는 스캐닝 모듈(105)에 의해 수집된 테스트 정보 또는 로깅 모듈(107)에 의해 저장된 테스트 정보를 분석할 수 있다

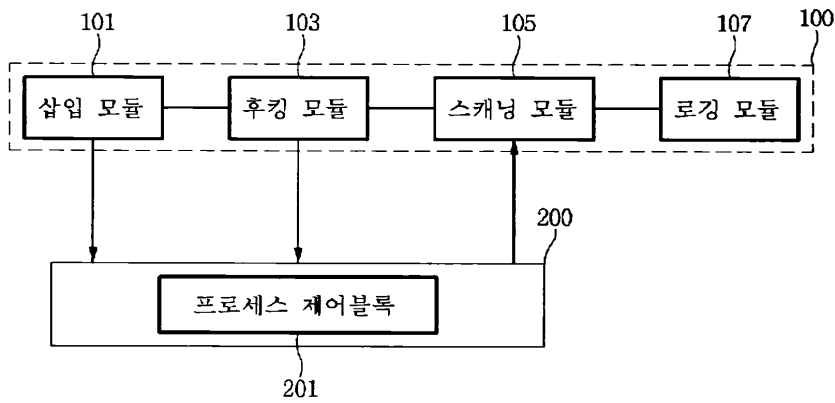
- [0069] 복수의 프로세스 사이의 통신에 대한 테스트를 수행하기 위해 이용되는 테스트 코드는 프로세스 사이의 메시지 전송, 메시지 큐, 공유 메모리 또는 동기화 객체와 관련된 결함 유형을 확인하기 위한 코드가 될 수 있다.
- [0070] 메시지 전송(Message Passing)과 관련된 결함 유형은 지정 시간 내 메시지 전송 실패(동기), 메시지 데드락(동기), 메시지 처리 지연(동기), 메시지 손실(동기), 메시지 변경(동기), 메시지 큐에 삽입 실패(비동기), 메시지 처리 지연(비동기), 메시지 손실(비동기), 또는 메시지 변경(비동기)으로 분류될 수 있다.
- [0071] 메시지 큐(Message Queue)와 관련된 결함 유형은 메시지 큐 생성 실패, 메시지 큐 닫기 실패, 메시지 큐로부터 읽기 실패, 또는 메시지 큐에 쓰기 실패로 분류될 수 있다.
- [0072] 공유 메모리(Shares Memory)와 관련된 결함 유형은 공유 메모리 생성 실패일 수 있다.
- [0073] 동기화 객체(Synchronization Object)와 관련된 결함 유형은 생성하지 않은 동기화 객체 사용, 생성한 동기화 객체 사용하지 않음, 또는 사용한 동기화 객체 해제 하지 않음으로 분류될 수 있다.
- [0074] 따라서, 각 결함 유형에 따라 테스트 코드가 변경된다.
- [0075] 상기한 바와 같이, 본 발명에 따른 시스템 테스트 장치(100)는 동작하고 있는 시스템(200)에 영향을 주지 않으면서 테스트 에이전트 삽입, 테스트 코드 후킹, 테스트 정보 수집을 실시할 수 있으며, 상기 시스템 테스트 장치(100)에 의한 시스템 테스트는 시스템에 포함된 프로세스 제어블록에 포함된 정보를 기초로 실시된다.
- [0076] 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.
- [0077] 본 발명은 상술한 일실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진자에 의해 다양한 변형실시가 가능하며, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어서는 안될 것이다.
- [0078] **산업상 이용가능성**
- [0079] 본 발명은 시스템이 실제 동작하는 동안 메모리 오류, 시스템 성능, 프로세스 사이의 통신을 테스트 할 수 있다.

도면

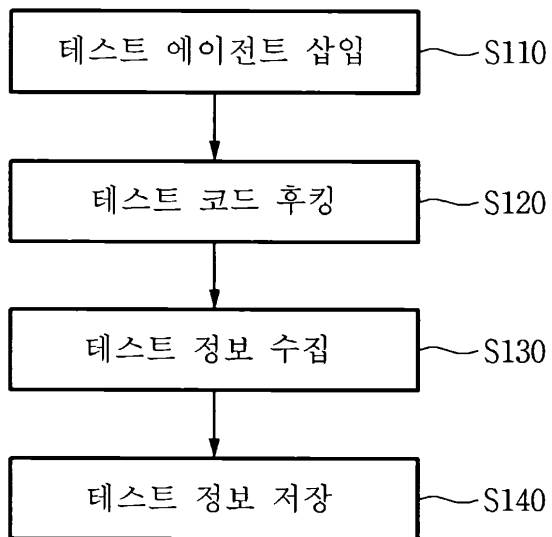
도면1



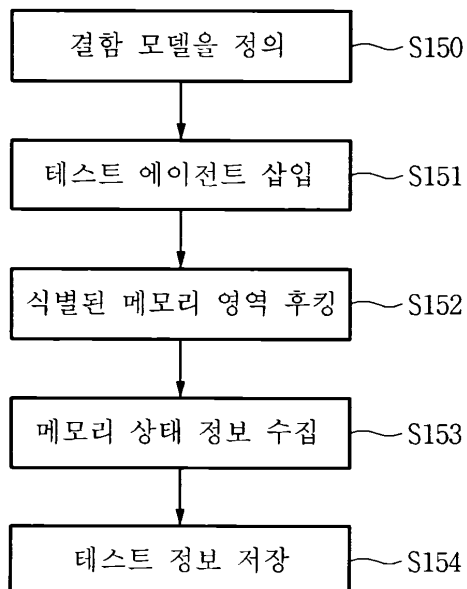
도면2



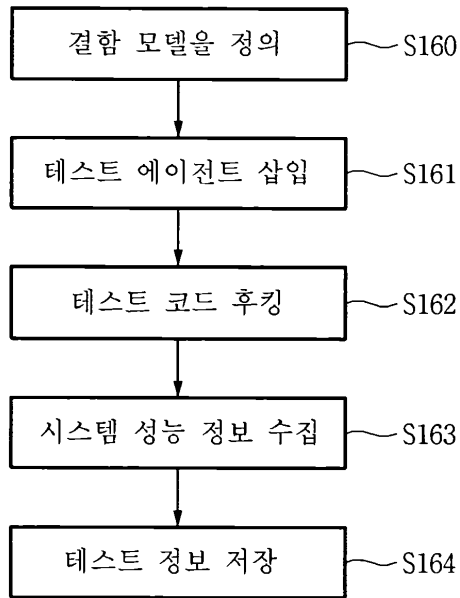
도면3



도면4a



도면4b



도면4c

