(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년11월05일
(11) 등록번호 10-2323621
(24) 등록일자 2021년11월02일

(51) 국제특허분류(Int. Cl.)
G06F 11/36 (2006.01) *G06F 11/263* (2006.01)
(52) CPC특허분류
G06F 11/3664 (2013.01)
G06F 11/263 (2013.01)
(21) 출원번호 10-2021-0057223
(22) 출원일자 2021년05월03일
심사청구일자 2021년05월03일
(56) 선행기술조사문헌
KR102209676 B1*
(뒷면에 계속)

(73) 특허권자
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
윤주범
서울특별시 송파구 충민로4길 19, 704동 401호(장지동, 송파파인타운7단지)
김주환
서울특별시 광진구 군자로 175-2, 304호(군자동)
(뒷면에 계속)
(74) 대리인
두호특허법인

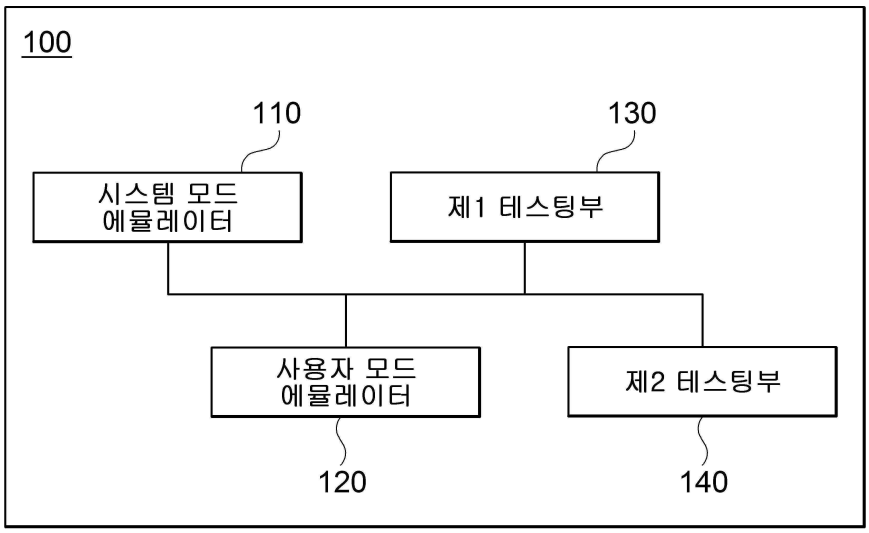
전체 청구항 수 : 총 10 항 심사관 : 김계준

(54) 발명의 명칭 **펌웨어 퍼징 장치 및 방법**

(57) 요 약

펌웨어(firmware) 퍼징(fuzzing) 장치 및 방법이 개시된다. 일 실시예에 따르면 펌웨어 퍼징 장치는 임의의 사물인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 시스템 모드 에뮬레이터; 상기 시스템 모드 에뮬레이션 환경에서 실행되는 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 사용자 모드 에뮬레이터; 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 대한 변이 기반 퍼징을 실행하는 제1 테스팅부; 및 상기 퍼징이 실행되는 도중 기 설정된 이벤트가 발생된 경우, 상기 시스템 모드 에뮬레이션 환경에서 상기 발생된 이벤트를 해결하기 위한 동작을 수행하는 제2 테스팅부를 포함하되, 상기 제1 테스팅부는, 상기 기 설정된 이벤트가 발생한 경우, 상기 퍼징의 실행을 중단하고, 상기 동작의 수행 결과에 기초하여 상기 퍼징의 실행을 재개한다.

대 표 도 - 도1



(52) CPC특허분류
G06F 11/3648 (2013.01)
G06F 9/4401 (2013.01)
G06F 9/45504 (2013.01)

(72) 발명자
유지현
서울특별시 광진구 군자로3길 18-2, 101호 (화양동)
이영우
경기도 구리시 경춘로288번길 39, 마동 410호(수택동)

(56) 선행기술조사문헌
DISCOVERING MEMORY CORRUPTIONS*
EFFECTIVE PROBING AND FUZZING FRAMEWORK*
VULNERABILITY-ORIENTED FUZZING OF IOT FIRMWARE*
US10599558 B1*
*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126109
과제번호	2018-0-01423-004
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	대학ICT연구센터육성지원사업
연구과제명	지능형 비행로봇 융합기술 연구
기 여 율	1/2
과제수행기관명	세종대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126138
과제번호	2020-0-01602-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	지능형 사이버 위협 대응 기술 개발 및 인력양성
기 여 율	1/2
과제수행기관명	숭실대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

임의의 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 시스템 모드 에뮬레이터;

상기 시스템 모드 에뮬레이션 환경에서 실행되는 상기 기기의 펌웨어(firmware)에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 사용자 모드 에뮬레이터;

상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 대한 변이 기반 퍼징(fuzzing)을 실행하는 제1 테스팅부; 및

상기 퍼징이 실행되는 도중 기 설정된 이벤트가 발생된 경우, 상기 시스템 모드 에뮬레이션 환경에서 상기 발생된 이벤트를 해결하기 위한 동작을 수행하는 제2 테스팅부를 포함하되,

상기 제1 테스팅부는,

상기 기 설정된 이벤트가 발생한 경우, 상기 퍼징의 실행을 중단하고, 상기 동작의 수행 결과에 기초하여 상기 퍼징의 실행을 재개하고,

상기 제2 테스팅부는,

사전 생성된 라이브러리(library)를 이용하여 상기 퍼징의 과정에서 실행될 하드웨어 의존성 함수를 처리하되, 상기 라이브러리는 상기 하드웨어 의존성 함수의 결과를 참(true)으로 반환하는, 펌웨어 퍼징 장치.

청구항 2

청구항 1에 있어서,

상기 제1 테스팅부는,

메모리에 저장된 상기 대상 프로세스에 대한 문자열 상수를 추출하고, 상기 문자열 상수에 기초하여 상기 퍼징을 실행하는, 펌웨어 퍼징 장치.

청구항 3

청구항 1에 있어서,

상기 제2 테스팅부는,

상기 퍼징이 실행되는 도중 페이지 폴트(page fault)가 발생하는 경우, 상기 페이지 폴트를 유발하는 명령어(instructor)를 상기 사용자 모드 에뮬레이션 환경에서 실행하여 상기 페이지 폴트를 처리하는, 펌웨어 퍼징 장치.

청구항 4

삭제

청구항 5

삭제

청구항 6

청구항 1에 있어서,

상기 제1 테스팅부는,

제1 입력 값에 따른 상기 퍼징의 코드 커버리지(code coverage) 증감 결과에 기초하여 제2 입력 값에 대한 퍼징을 수행하는, 펌웨어 퍼징 장치.

청구항 7

청구항 6에 있어서,

상기 제1 테스팅부는,

상기 제1 입력 값에 따른 상기 퍼징의 코드 커버리지가 감소되는 경우, 상기 사용자 모드 에뮬레이션 환경을 초기화시킨 후, 상기 제2 입력 값에 대한 퍼징을 수행하는, 펌웨어 퍼징 장치.

청구항 8

임의의 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 단계;

상기 시스템 모드 에뮬레이션 환경에서 실행되는 상기 기기의 펌웨어(firmware)에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 단계;

상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 대한 변이 기반 퍼징(fuzzing)을 실행하는 단계; 및

상기 퍼징이 실행되는 도중 기 설정된 이벤트가 발생된 경우, 상기 시스템 모드 에뮬레이션 환경에서 상기 발생된 이벤트를 해결하기 위한 동작을 수행하는 단계를 포함하되,

상기 퍼징을 실행하는 단계는,

상기 기 설정된 이벤트가 발생한 경우, 상기 퍼징의 실행을 중단하고, 상기 동작의 수행 결과에 기초하여 상기 퍼징의 실행을 재개하고,

상기 동작을 수행하는 단계는,

사전 생성된 라이브러리(library)를 이용하여 상기 퍼징의 과정에서 실행될 하드웨어 의존성 함수를 처리하되, 상기 라이브러리는 상기 하드웨어 의존성 함수의 결과를 참(true)으로 반환하는, 펌웨어 퍼징 방법.

청구항 9

청구항 8에 있어서,

상기 퍼징을 실행하는 단계는,

메모리에 저장된 상기 대상 프로세스에 대한 문자열 상수를 추출하고, 상기 문자열 상수에 기초하여 상기 퍼징을 실행하는, 펌웨어 퍼징 방법.

청구항 10

청구항 8에 있어서,

상기 동작을 수행하는 단계는,

상기 퍼징이 실행되는 도중 페이지 폴트(page fault)가 발생하는 경우, 상기 페이지 폴트를 유발하는 명령어(instructor)를 상기 사용자 모드 에뮬레이션 환경에서 실행하여 상기 페이지 폴트를 처리하는, 펌웨어 퍼징 방법.

청구항 11

삭제

청구항 12

삭제

청구항 13

청구항 8에 있어서,

상기 퍼징을 수행하는 단계는,

제1 입력 값에 따른 상기 퍼징의 코드 커버리지(code coverage) 증감 결과에 기초하여 제2 입력 값에 대한 퍼징을 수행하는, 펌웨어 퍼징 방법.

청구항 14

청구항 13에 있어서,

상기 퍼징을 수행하는 단계는,

상기 제1 입력 값에 따른 상기 퍼징의 코드 커버리지가 감소되는 경우, 상기 사용자 모드 에뮬레이션 환경을 초기화시킨 후, 상기 제2 입력 값에 대한 퍼징을 수행하는, 펌웨어 퍼징 방법.

발명의 설명

기 술 분 야

[0001] 개시되는 실시예들은 펌웨어(firmware)에 대한 변이 기반 퍼징(fuzzing)을 수행하는 기술과 관련된다.

배 경 기 술

[0002] 사물 인터넷(IoT; Internet of Things)(이하, IoT) 기술의 발전으로 인해 다양한 임베디드 디바이스가 개발되고, 그에 따른 다양한 펌웨어(firmware)가 개발되고 있다. 이에, IoT 장치 내 설치된 모든 펌웨어에 대해 수동으로 취약점을 분석하기에는 물리적인 한계를 갖는다.

[0003] 상술한 한계를 극복하고자, 에뮬레이션 기반의 퍼징(fuzzing)과 관련된 많은 연구들이 진행되었다. 여기서, 에뮬레이션에는 가상 머신과 동일한 기능을 수행하는 시스템 모드 에뮬레이션과 대상 프로세스를 디버깅(debugging)하는 사용자 모드 에뮬레이션이 존재한다.

[0004] 다만, 시스템 모드 에뮬레이션은 모든 하드웨어를 소프트웨어로 구현했기에, 성능 오버헤드로 인해 낮은 처리량에 대한 한계를 가지며, 사용자 모드 에뮬레이션 은 하드웨어의 결핍으로 인해 호환성이 낮다는 한계를 가진다.

[0005] 또한, 일부 IoT 장치들은 하드웨어 의존성 함수에 의해 주변 장치에 접근을 시도할 수 있는데, 이러한 주변 장치는 시스템 모드 에뮬레이션으로 구현될 수 없으므로 에뮬레이션의 충돌을 유발할 수 있다. 결국, 일부 IoT 장치들의 네트워크와 관련된 프로세스들은 에뮬레이션 환경에서 실행되는 퍼징에 있어서, 구조화된 입력 값만을 받을 수 있다는 한계가 존재한다. 이에, 상술한 한계들을 해결할 수 있는 새로운 펌웨어 취약점 탐지 방법이 필요한 실정이다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 대한민국 공개특허공보 제10-2020-0080541호(2020.07.07. 등록)

발명의 내용

해결하려는 과제

[0007] 개시되는 실시예들은 펌웨어(firmware)에 대한 변이 기반 퍼징(fuzzing)을 수행하기 위한 장치 및 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0008] 일 실시예에 따른 펌웨어(firmware) 퍼징(fuzzing) 장치는 임의의 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 시스템 모드 에뮬레이터; 상기 시스템 모드 에뮬레이션 환경에서 실행되는 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 사용자 모드 에뮬레이터; 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 대한 변이 기반 퍼징을 실행하는 제1 테스팅부; 및 상기 퍼징이 실행되는 도중 기 설정된 이벤트가 발생된 경우, 상기 시스템 모드 에뮬레이션 환경에서 상기 발생된 이벤트를 해결하기 위한 동작을 수행하는 제2 테스팅부를 포함하되, 상기 제1 테스팅부는, 상기 기 설정된 이벤트가 발생한 경우, 상기 퍼징의 실행을 중단하고, 상기 동작의 수행 결과에 기초하여 상기 퍼징의 실행을 재개한다.

[0009] 상기 제1 테스팅부는, 메모리에 저장된 상기 대상 프로세스에 대한 문자열 상수를 추출하고, 상기 문자열 상수에 기초하여 상기 퍼징을 실행할 수 있다.

[0010] 상기 제2 테스팅부는, 상기 퍼징이 실행되는 도중 페이지 폴트(page fault)가 발생하는 경우, 상기 페이지 폴트를 유발하는 명령어(instructor)를 상기 사용자 모드 에뮬레이션 환경에서 실행하여 상기 페이지 폴트를 처리할 수 있다.

[0011] 상기 제2 테스팅부는, 사전 생성된 라이브러리(library)를 이용하여 상기 퍼징의 과정에서 실행될 하드웨어 의존성 함수를 처리할 수 있다.

[0012] 상기 라이브러리는, 상기 하드웨어 의존성 함수의 결과를 참(true)로 반환할 수 있다.

[0013] 상기 제1 테스팅부는, 제1 입력 값에 따른 상기 퍼징의 코드 커버리지(code coverage) 증감 결과에 기초하여 제2 입력 값에 대한 퍼징을 수행할 수 있다.

[0014] 상기 제1 테스팅부는, 상기 제1 입력 값에 따른 상기 퍼징의 코드 커버리지가 감소되는 경우, 상기 사용자 모드 에뮬레이션 환경을 초기화시킨 후, 상기 제2 입력 값에 대한 퍼징을 수행할 수 있다.

[0015] 일 실시예에 따른 펌웨어 퍼징 방법은 임의의 사물 인터넷(IoT; Internet of Things) 기기에 시스템 모드 에뮬레이션 환경을 제공하는 단계; 상기 시스템 모드 에뮬레이션 환경에서 실행되는 상기 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공하는 단계; 상기 사용자 모드 에뮬레이션 환경에서 상기 대상 프로세스에 대한변이 기반 퍼징을 실행하는 단계; 및 상기 퍼징이 실행되는 도중 기 설정된 이벤트가 발생된 경우, 상기 시스템 모드 에뮬레이션 환경에서 상기 발생된 이벤트를 해결하기 위한 동작을 수행하는 단계를 포함하되, 상기 퍼징을 실행하는 단계는, 상기 기 설정된 이벤트가 발생한 경우, 상기 퍼징의 실행을 중단하고, 상기 동작의 수행 결과에 기초하여 상기 퍼징의 실행을 재개한다.

[0016] 상기 퍼징을 실행하는 단계는, 메모리에 저장된 상기 대상 프로세스에 대한 문자열 상수를 추출하고, 상기 문자열 상수에 기초하여 상기 퍼징을 실행할 수 있다.

[0017] 상기 동작을 수행하는 단계는, 상기 퍼징이 실행되는 도중 페이지 폴트(page fault)가 발생하는 경우, 상기 페이지 폴트를 유발하는 명령어(instructor)를 상기 사용자 모드 에뮬레이션 환경에서 실행하여 상기 페이지 폴트를 처리할 수 있다.

[0018] 상기 동작을 수행하는 단계는, 사전 생성된 라이브러리(library)를 이용하여 상기 퍼징의 과정에서 실행될 하드웨어 의존성 함수를 처리할 수 있다.

[0019] 상기 라이브러리는, 상기 하드웨어 의존성 함수의 결과를 참(true)로 반환할 수 있다.

[0020] 상기 퍼징을 수행하는 단계는, 제1 입력 값에 따른 상기 퍼징의 코드 커버리지(code coverage) 증감 결과에 기초하여 제2 입력 값에 대한 퍼징을 수행할 수 있다.

[0021] 상기 퍼징을 수행하는 단계는, 상기 제1 입력 값에 따른 상기 퍼징의 코드 커버리지가 감소되는 경우, 상기 사용자 모드 에뮬레이션 환경을 초기화시킨 후, 상기 제2 입력 값에 대한 퍼징을 수행할 수 있다.

발명의 효과

[0022] 개시되는 실시예들에 따르면, 시스템 모드 에뮬레이터와 사용자 모드 에뮬레이터를 복합적으로 사용함으로써 호환성과 동시에 처리 속도가 향상된 에뮬레이션 기반의 퍼징 기법을 제공할 수 있다.

[0023] 개시되는 실시예들에 따르면, 문자열 상수를 이용하여 퍼징을 수행함으로써, 입력 값에 대한 제한 없는 퍼징 기법을 제공할 수 있다.

[0024] 개시되는 실시예들에 따르면, 퍼징 수행 시 제1 테스팅부가 해결할 수 없는 인터럽트(interrupt)를 제2 테스팅부를 통해 해결함으로써 안정적인 퍼징을 수행할 수 있다.

[0025] 개시되는 실시예들에 따르면, 퍼징 결과를 고려하여 선별된 입력 값을 기반으로 퍼징을 수행함으로써, 코드 커버리지(code coverage)를 넓힐 수 있다.

도면의 간단한 설명

[0026] 도 1은 일 실시예에 따른 펌웨어(firmware) 퍼징(fuzzing) 장치의 블록도

도 2는 일 실시예에 따른 펌웨어 퍼징 방법을 설명하기 위한 흐름도

도 3은 추가적인 실시예에 따른 데이터 관리 방법을 설명하기 위한 흐름도

도 4는 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

발명을 실시하기 위한 구체적인 내용

[0027] 이하, 도면을 참조하여 일 실시예의 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서 기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는 예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0028] 일 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 일 실시예의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 일 실시예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 성분들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 성분, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0029] 도 1은 일 실시예에 따른 펌웨어(firmware) 퍼징(fuzzing) 장치(100)를 설명하기 위한 구성도이다.

[0030] 도 1을 참조하면, 일 실시예에 따른 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이터(110), 사용자 모드 에뮬레이터(120), 제1 테스팅부(130) 및 제2 테스팅부(140)를 포함한다.

[0031] 이때, 시스템 모드 에뮬레이터(110), 사용자 모드 에뮬레이터(120), 제1 테스팅부(130) 및 제2 테스팅부(140)는 각각 물리적으로 구분된 하나 이상의 장치를 이용하여 구현되거나, 하나 이상의 하드웨어 프로세서 또는 하나 이상의 하드웨어 프로세서 및 소프트웨어의 결합에 의해 구현될 수 있으며, 도시된 예와 달리 구체적 동작에 있어 명확히 구분되지 않을 수 있다.

[0032] 시스템 모드 에뮬레이터(110)는 임의의 사물 인터넷(IoT; Internet of Things)(이하, IoT) 기기에 설치된 펌웨어에 시스템 모드 에뮬레이션(emulation) 환경을 제공한다. 여기서, 시스템 모드 에뮬레이션 환경이란, IoT 기기와 관련된 시스템 전체에 대한 에뮬레이팅(emulating)을 수행할 수 있는 환경을 의미한다.

[0033] 사용자 모드 에뮬레이터(120)는 시스템 모드 에뮬레이션 환경에서 실행되는 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공한다. 여기서, 사용자 모드 에뮬레이션 환경이란, 대상 프로세스에 대한 에뮬레이팅을 수행할 수 있는 환경을 의미한다. 이때, 대상 프로세스란, 펌웨어를

실행함에 따라 발생되는 하나 이상의 프로세스 중 퍼징 수행의 타겟이 되는 프로세스를 의미한다.

[0034] 제1 테스팅부(130)는 사용자 모드 에뮬레이션 환경에서 대상 프로세스에 대한 변이 기반 퍼징을 실행한다.

[0035] 변이 기반 퍼징이란, 퍼징 수행 시 기 마련된 시드 파일(seed file)을 변형하여 테스트 케이스를 생성하는 퍼징 기법을 지칭한다.

[0036] 일 실시예에 따르면, 제1 테스팅부(130)는 대상 프로세스에 대한 문자열 상수를 추출하고, 추출된 문자열 상수에 기초하여 퍼징을 수행할 수 있다. 이를 통해, 제1 테스팅부(130)는 입력 값 형식에 구애 받지 않고 퍼징을 수행할 수 있다. 한편, 이때 문자열 상수는 정적 분석을 통해 추출될 수 있다.

[0037] 한편, 일 실시예에 따르면, 제1 테스팅부(130)는 제1 입력 값에 따른 퍼징의 코드 커버리지(code coverage) 증감 결과에 기초하여 제2 입력 값에 대한 퍼징을 수행할 수 있다.

[0038] 다시 말해, 제1 테스팅부(130)는, 제1 입력 값에 따른 퍼징의 코드 커버리지가 감소되는 경우, 사용자 모드 에뮬레이션 환경을 초기화시킨 후, 높은 처리량을 위해 사용자 모드 에뮬레이션 환경에서 코드 커버리지를 넓힐 수 있도록 제2 입력 값에 대한 퍼징을 다시 수행할 수 있다.

[0039] 이를 통해, 일 실시예에 따르면, 제1 테스팅부(130)는 코드 커버리지가 증가될 수 있도록 퍼징을 반복 수행하는 그레이 박스(grey box) 방식의 퍼징을 수행할 수 있다.

[0040] 제2 테스팅부(140)는 퍼징의 실행 중 기 설정된 이벤트가 발생된 경우, 발생된 이벤트를 해결하기 위한 동작을 시스템 모드 에뮬레이션 환경에서 수행한다.

[0041] 일 실시예에 따르면, 제2 테스팅부(140)는 기 설정된 이벤트가 발생되는 경우로서, 퍼징이 실행되는 도중 페이지 폴트(page fault)가 발생하는 경우, 페이지 폴트를 해결하기 위한 동작으로 페이지 폴트를 유발한 명령어를 실행하여 페이지 폴트를 처리할 수 있다.

[0042] 다른 실시예에 따르면, 제2 테스팅부(140)는 기 설정된 이벤트가 발생되는 경우로서, 퍼징이 실행되는 도중 하드웨어 의존성 함수의 실행이 요구되는 경우, 사전 생성된 라이브러리(library)를 이용하여 하드웨어 의존성 함수를 처리할 수 있다.

[0043] 여기서, 하드웨어 의존성 함수는 IoT 기기가 IoT 기기의 주변 기기로 접근을 가능하게 하는 함수를 의미한다.

[0044] 예를 들어, IoT 기기가 주변 기기인 NVRAM 장치로부터 config 파일을 읽는 함수를 nvram_get()라고 가정하면, 이때 하드웨어 의존 함수는 nvram_get()일 수 있다.

[0045] 다시 말해, 제1 테스팅부(130)가 퍼징을 수행하는 도중 nvram_get() 함수의 실행이 요구되는 경우, 제2 테스팅부(140)는 라이브러리를 이용하여 nvram_get() 함수를 처리할 수 있다.

[0046] 이때, 라이브러리란, 하드웨어 의존성 함수의 결과를 참(true)로 반환할 수 있도록 함수나 데이터들을 미리 만들어 모아 놓은 집합체이다.

[0047] 즉, 제2 테스팅부(140)는 nvram_get() 함수의 실행이 요구될 때 라이브러리를 통해 nvram_get() 함수의 결과 값을 참으로 반환함에 따라 NVRAM 장치로 접근하지 않으면서도 nvram_get() 함수를 처리할 수 있다.

[0048] 이를 통해, 제2 테스팅부(140)는 주변 장치에 대해 에뮬레이션을 수행할 수 없는 에뮬레이터의 한계를 해결할 수 있다.

[0049] 도 2는 일 실시예에 따른 펌웨어 퍼징 방법을 설명하기 위한 흐름도이다.

[0050] 도 2에 도시된 방법은 도 1에서 도시된 펌웨어 퍼징 장치(100)에 의해 수행될 수 있다.

[0051] 도 2를 참조하면, 펌웨어 퍼징 장치(100)는 임의의 사물 인터넷 기기에 시스템 모드 에뮬레이션 환경을 제공한다(210).

[0052] 이후, 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이션 환경에서 실행되는 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공한다(220).

[0053] 이후, 펌웨어 퍼징 장치(100)는 사용자 모드 에뮬레이션 환경에서 대상 프로세스에 대한 변이 기반 퍼징을 실행한다(230).

[0054] 이후, 펌웨어 퍼징 장치(100)는 퍼징이 실행되는 도중 기 설정된 이벤트가 발생되었는지 여부를 판단한다(240).

[0055] 이때, 펌웨어 퍼징 장치(100)는 펌웨어 퍼징 장치(100)는 기 설정된 이벤트가 발생되었다고 판단한 경우, 실행 중인 퍼징을 중단한다(250).

[0056] 이후, 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이션 환경에서 발생된 이벤트를 해결하기 위한 동작을 수행한다(260).

[0057] 이후, 펌웨어 퍼징 장치(100)는 동작의 수행 결과에 기초하여 퍼징의 실행을 재개한다(270).

[0058] 이후, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되었는지 여부를 판단한다(280).

[0059] 이때, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되지 않았다고 판단된 경우, 퍼징 종료 조건이 만족될 때까지 단계 240 내지 270을 반복 수행한다.

[0060] 한편, 퍼징 종료 조건은 예를 들어, 퍼징이 실행된 횟수일 수 있으나 실시예에 따라 다양하게 설정될 수 있다.

[0061] 도 3은 추가적인 실시예에 따른 데이터 관리 방법을 설명하기 위한 흐름도이다.

[0062] 도 3에 도시된 방법은 도 1에서 도시된 펌웨어 퍼징 장치(100)에 의해 수행될 수 있다.

[0063] 도 3을 참조하면, 펌웨어 퍼징 장치(100)는 임의의 사물 인터넷 기기에 시스템 모드 에뮬레이션 환경을 제공한다(301).

[0064] 이후, 펌웨어 퍼징 장치(100)는 시스템 모드 에뮬레이션 환경에서 실행되는 기기의 펌웨어에 대한 하나 이상의 프로세스 중 대상 프로세스에 사용자 모드 에뮬레이션 환경을 제공한다(302).

[0065] 이후, 펌웨어 퍼징 장치(100)는 사용자 모드 에뮬레이션 환경에서 대상 프로세스에 대한 변이 기반 퍼징을 실행한다(303).

[0066] 이후, 펌웨어 퍼징 장치(100)는 퍼징이 실행되는 도중 페이지 폴트가 발생되었는지 여부를 판단한다(304).

[0067] 이때, 펌웨어 퍼징 장치(100)는 페이지 폴트가 발생되었다고 판단한 경우, 페이지 폴트를 유발한 명령어를 실행한다(305).

[0068] 이후, 펌웨어 퍼징 장치(100)는 페이지 폴트가 처리될 당시 대상 프로세스의 메모리 상태를 저장한다(306).

[0069] 이후, 펌웨어 퍼징 장치(100)는 페이지 폴트가 처리될 당시 대상 프로세스의 메모리 상태에 기초하여 퍼징을 재개한다(307).

[0070] 이후, 펌웨어 퍼징 장치(100)는 퍼징이 실행되는 도중 하드웨어 의존성 함수의 실행이 요구되는지 여부를 판단한다(308).

[0071] 이때, 펌웨어 퍼징 장치(100)는 하드웨어 의존성 함수의 실행이 요구되는 경우, 시스템 모드 에뮬레이션 환경에서 라이브러리를 통해 하드웨어 의존성 함수를 처리한다(309).

[0072] 이후, 펌웨어 퍼징 장치(100)는 하드웨어 의존성 함수가 처리될 당시 대상 프로세스의 메모리 상태를 저장한다(310).

[0073] 이후, 펌웨어 퍼징 장치(100)는 하드웨어 의존성 함수가 처리될 당시 대상 프로세스의 메모리 상태에 기초하여 퍼징을 재개한다(311).

[0074] 이후, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되었는지 여부를 판단한다(312).

[0075] 이때, 펌웨어 퍼징 장치(100)는 펌웨어에 대한 퍼징 종료 조건이 만족되지 않았다고 판단된 경우, 퍼징 종료 조건이 만족될 때까지 단계 303 내지 311을 반복 수행한다.

[0076] 한편, 퍼징 종료 조건은 예를 들어, 퍼징이 실행된 횟수일 수 있으나 실시예에 따라 다양하게 설정될 수 있다.

[0077] 한편, 도 2 및 도 3에 도시된 실시예는 복수 개의 단계로 나누어 기재되었으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0078] 도 4는 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

[0079] 도 4는 일 실시예에 따르면 컴퓨팅 장치(12)를 포함하는 컴퓨팅 환경(10)을 예시하여 설명하기 위한 블록도이다. 도시된 실시예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있

고, 이하에 기술되지 않은 것 이외에도 추가적인 컴포넌트를 포함할 수 있다.

[0080] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 일 실시예에서, 컴퓨팅 장치(12)는 펌웨어 퍼징 장치(100)에 포함된 하나 이상의 컴포넌트일 수 있다.

[0081] 컴퓨팅 장치(12)는 적어도 하나의 프로그램(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로그램(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로그램(14)는 컴퓨터 판독 가능 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로그램(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0082] 컴퓨터 판독 가능 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능 저장 매체(16)에 저장된 프로그램(20)은 프로그램(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능 저장 매체(16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨팅 장치(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0083] 통신 버스(18)는 프로그램(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0084] 컴퓨팅 장치(12)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(22) 및 하나 이상의 네트워크 통신 인터페이스(26)를 포함할 수 있다. 입출력 인터페이스(22) 및 네트워크 통신 인터페이스(26)는 통신 버스(18)에 연결된다. 입출력 장치(24)는 입출력 인터페이스(22)를 통해 컴퓨팅 장치(12)의 다른 컴포넌트들에 연결될 수 있다. 예시적인 입출력 장치(24)는 포인팅 장치(마우스 또는 트랙패드 등), 키보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출력 장치를 포함할 수 있다. 예시적인 입출력 장치(24)는 컴퓨팅 장치(12)를 구성하는 일 컴포넌트로서 컴퓨팅 장치(12)의 내부에 포함될 수도 있고, 컴퓨팅 장치(12)와는 구별되는 별개의 장치로 컴퓨팅 장치(12)와 연결될 수도 있다.

[0085] 이상에서 대표적인 실시예를 통하여 본 발명에 대하여 상세하게 설명하였으나, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 전술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리범위는 설명된 실시예에 국한되어 정해져서는 안 되며, 후술하는 청구범위뿐만 아니라 이 청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

[0086] 10: 컴퓨팅 환경

12: 컴퓨팅 장치

14: 프로그램

16: 컴퓨터 판독 가능 저장 매체

18: 통신 버스

20: 프로그램

22: 입출력 인터페이스

24: 입출력 장치

26: 네트워크 통신 인터페이스

100: 펌웨어(firmware) 퍼징(fuzzing) 장치

110: 시스템 모드 에뮬레이터

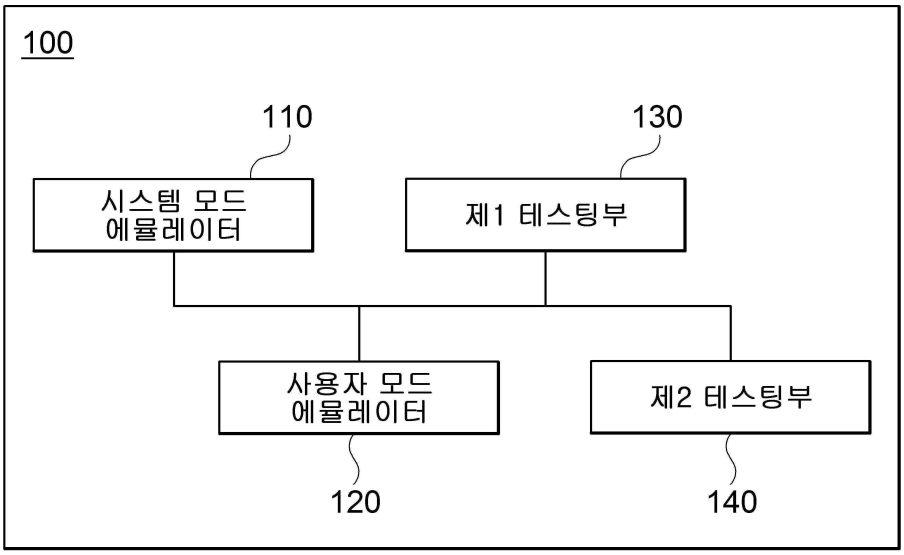
120: 사용자 모드 에뮬레이터

130: 제1 테스팅부

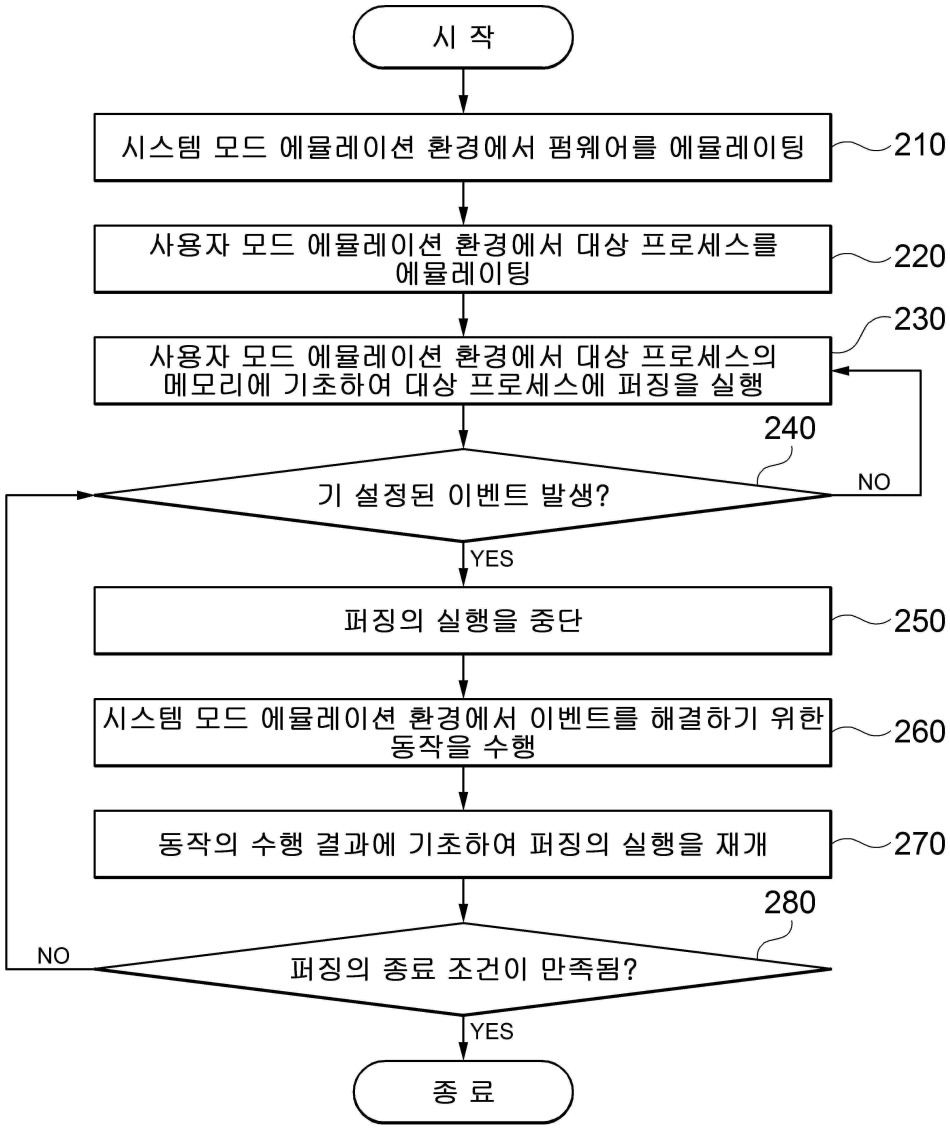
140: 제2 테스팅부

도면

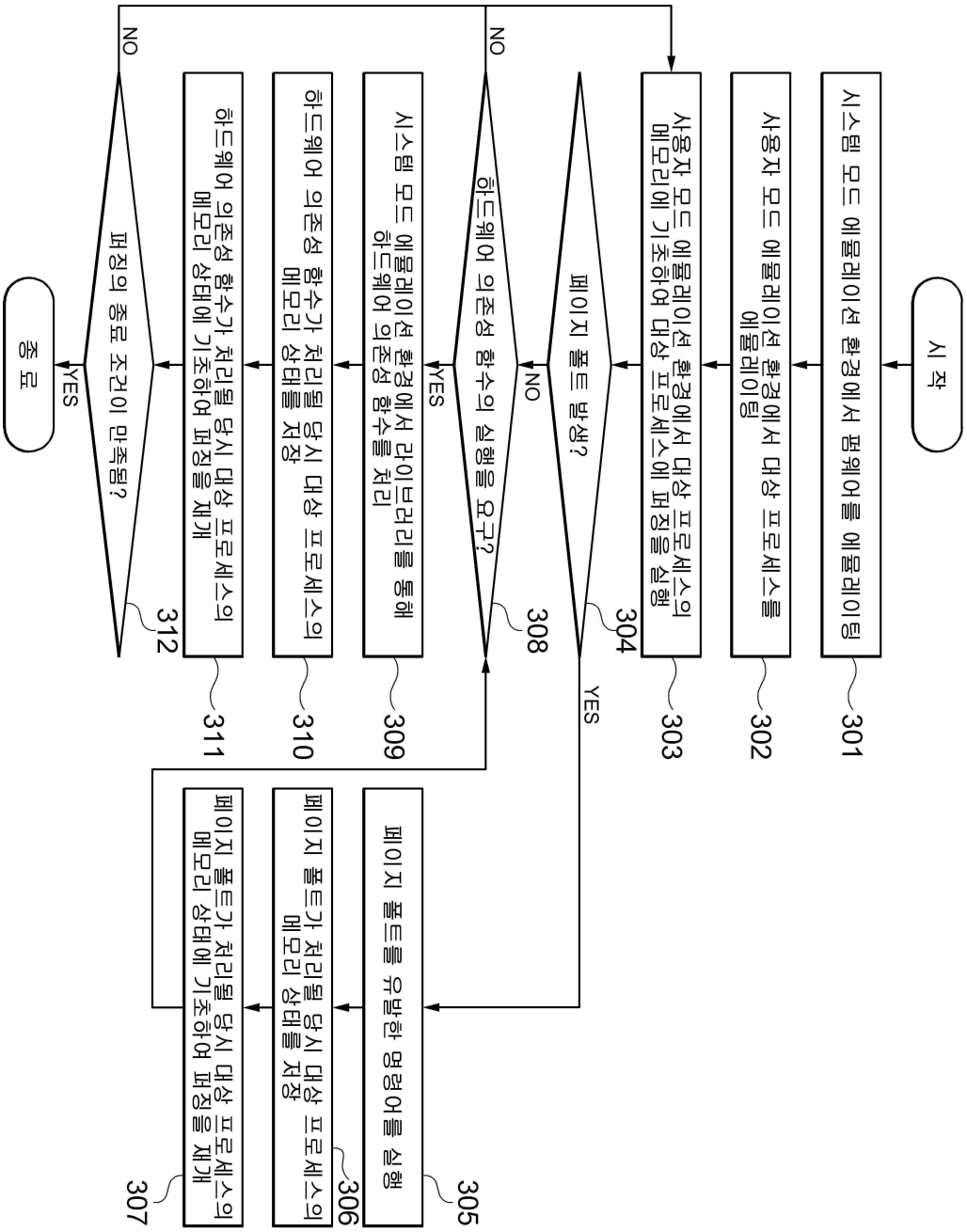
도면1



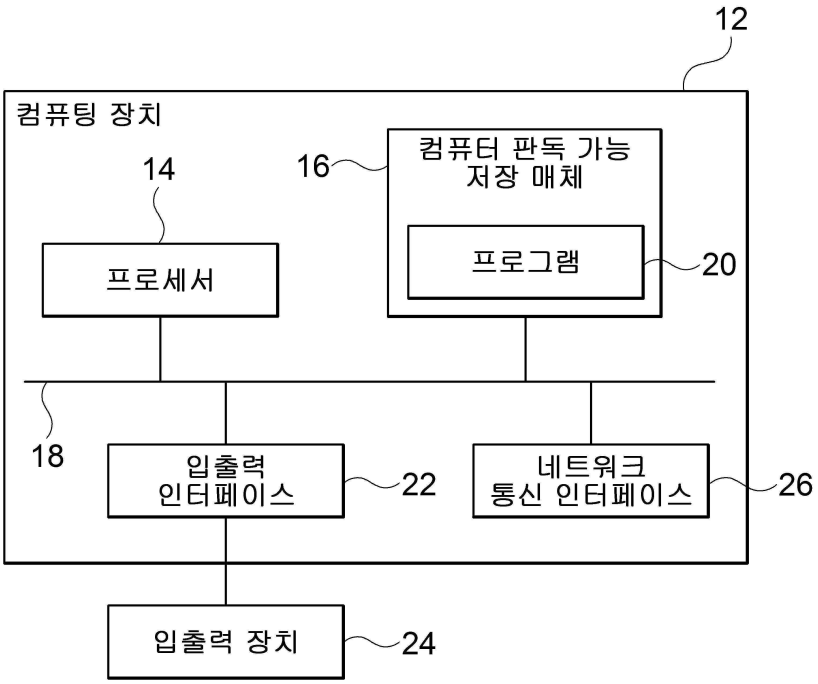
도면2


시 작
시스템 모드 에뮬레이션 환경에서 펌웨어를 에뮬레이팅
210
사용자 모드 에뮬레이션 환경에서 대상 프로세스를 에뮬레이팅
220
230
사용자 모드 에뮬레이션 환경에서 대상 프로세스의 메모리에 기초하여 대상 프로세스에 퍼징을 실행
240
기 설정된 이벤트 발생?
NO
YES
퍼징의 실행을 중단
250
시스템 모드 에뮬레이션 환경에서 이벤트를 해결하기 위한 동작을 수행
260
동작의 수행 결과에 기초하여 퍼징의 실행을 재개
270
280
퍼징의 종료 조건이 만족됨?
NO
YES
종 료

도면3


시 작
시스템 모드 에뮬레이션 환경에서 펌웨어를 에뮬레이팅
301
사용자 모드 에뮬레이션 환경에서 대상 프로세스를 에뮬레이팅
302
사용자 모드 에뮬레이션 환경에서 대상 프로세스의 메모리에 기초하여 대상 프로세스에 퍼징을 실행
303
페이지 폴트 발생?
304
YES
NO
하드웨어 의존성 함수의 실행을 요구?
308
YES
NO
시스템 모드 에뮬레이션 환경에서 라이브러리를 통해 하드웨어 의존성 함수를 처리
309
하드웨어 의존성 함수가 처리될 당시 대상 프로세스의 메모리 상태를 저장
310
하드웨어 의존성 함수가 처리될 당시 대상 프로세스의 메모리 상태에 기초하여 퍼징을 재개
311
퍼징의 종료 조건이 만족됨?
312
YES
NO
종 료
페이지 폴트를 유발한 명령어를 실행
305
페이지 폴트가 처리될 당시 대상 프로세스의 메모리 상태를 저장
306
페이지 폴트가 처리될 당시 대상 프로세스의 메모리 상태에 기초하여 퍼징을 재개
307

도면4


10
12
컴퓨팅 장치
14
프로세서
16
컴퓨터 판독 가능
저장 매체
프로그램
20
18
입출력
인터페이스
22
네트워크
통신 인터페이스
26
입출력 장치
24