



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2021년05월18일

(11) 등록번호 10-2254159

(24) 등록일자 2021년05월13일

(51) 국제특허분류(Int. Cl.)

G06F 11/30 (2006.01) G06F 11/10 (2006.01)

G06F 12/02 (2018.01) G06F 13/24 (2006.01)

G06F 3/06 (2006.01)

(52) CPC특허분류

G06F 11/3037 (2013.01)

G06F 11/1016 (2013.01)

(21) 출원번호 10-2020-0171585

(22) 출원일자 2020년12월09일

심사청구일자 2020년12월09일

(56) 선행기술조사문헌

KR1020130131026 A\*

(뒷면에 계속)

(73) 특허권자

연세대학교 산학협력단

서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)

(72) 발명자

권태경

서울특별시 강남구 선릉로 221, 410동 1602호(도곡동, 도곡렉슬아파트)

조민기

서울특별시 서대문구 증가로 26, 302호(연희동)  
(뒷면에 계속)

(74) 대리인

특허법인우인

전체 청구항 수 : 총 13 항

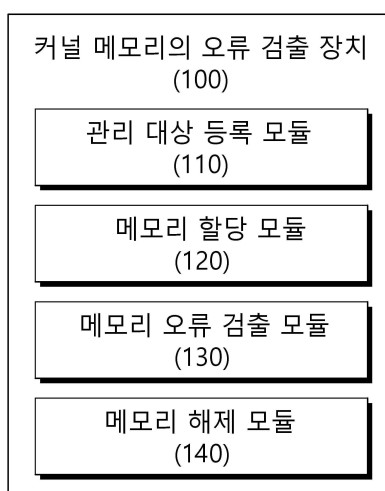
심사관 : 김계준

(54) 발명의 명칭 운영체제 커널 메모리의 실시간 오류 검출 방법

## (57) 요약

본 실시예들은 소스 코드 없이 실시간으로 대상 프로세스만을 선택하고 대상 프로세스에 대해서만 운영체제 커널의 메모리 오류를 검사하여 커널의 성능 영향을 최소화할 수 있는 커널 메모리의 실시간 오류 검출 방법 및 오류 검출 장치를 제공한다.

대표도 - 도1



(52) CPC특허분류

*G06F 11/1024* (2013.01)

*G06F 12/0238* (2013.01)

*G06F 13/24* (2013.01)

*G06F 3/0653* (2013.01)

(72) 발명자

**안도현**

대구광역시 수성구 달구벌대로 3280-1, 204동 601호(신매동, 시지 효성 백년가약 2단지)

**여기수**

부산광역시 부산진구 동평로 409, 1동 703호(양정동, 백조아파트)

(56) 선행기술조사문헌

KR102044075 B1\*

KR1020190135337 A

KR100645983 B1

JP2014067214 A

\*는 심사관에 의하여 인용된 문헌

이 발명을 지원한 국가연구개발사업

과제고유번호 1711103203

과제번호 2018-0-00513-003

부처명 과학기술정보통신부

과제관리(전문)기관명 정보통신기획평가원

연구사업명 정보통신방송연구개발사업

연구과제명 기계학습을 활용한 UNIX 기반 커널 취약점 탐지 자동화 연구(3/5)

기 여 율 1/1

과제수행기관명 연세대학교 산학협력단

연구기간 2020.01.01 ~ 2020.12.31

## 명세서

### 청구범위

#### 청구항 1

커널 메모리의 실시간 오류 검출 방법에 있어서,  
 관리 대상 프로세스를 등록하는 단계;  
 상기 관리 대상 프로세스를 기준으로 상기 커널의 메모리 주소 공간에 메모리 영역을 할당하는 단계;  
 상기 커널의 인터럽트 핸들러를 이용하여 상기 메모리 영역에 대해서 메모리 오류를 검출하는 단계; 및  
 상기 메모리 영역에 대해서 할당 해제하는 단계를 포함하며,  
 상기 메모리 영역을 할당하는 단계는,  
 실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하지 않으면 상기 커널의 메모리 주소 공간의 일반 메모리 영역에 메모리를 할당하는 단계; 및  
 상기 실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하면 상기 커널의 메모리 주소 공간의 검출 대상 메모리 영역에 메모리를 할당하는 단계를 포함하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 2

제1항에 있어서,  
 복수의 프로세스가 동시에 실행되는 상황에서, 상기 복수의 프로세스가 상기 커널의 메모리 주소 공간을 공유하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 3

삭제

#### 청구항 4

제1항에 있어서,  
 상기 메모리 영역을 할당하는 단계는,  
 상기 검출 대상 메모리 영역에 레드존을 설정하는 단계; 및  
 상기 커널의 메모리 주소 공간에 상기 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하는 단계를 포함하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 5

제4항에 있어서,  
 상기 메모리 영역을 할당하는 단계는,  
 상기 커널의 메모리 주소 공간의 가상 주소에 대한 페이지 접근 권한을 변경하는 단계를 포함하고,  
 상기 메모리 오류를 검출하는 단계는,  
 상기 커널의 인터럽트 핸들러에 의한 메모리 오류 검사를 통해 접근을 시도하는 가상 주소가 상기 검출 대상 메모리 영역인지 상기 일반 메모리 영역인지 판별하는 단계를 포함하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 6

제5항에 있어서,

상기 메모리 오류를 검출하는 단계는,

상기 가상 주소가 상기 검출 대상 메모리 영역에 해당하면,

상기 새도우 메모리 영역을 검사하여 메모리 오류를 확인하고, 오류에 해당하면 제1 상세 오류 유형 플래그를 갖는 오류 정보를 출력하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 7

제5항에 있어서,

상기 가상 주소가 상기 검출 대상 메모리 영역에 해당하면,

상기 새도우 메모리 영역을 검사하여 메모리 오류를 확인하고, 오류에 해당하지 않으면 상기 가상 주소에 대한 페이지 접근 권한을 가능으로 변경한 후 단일 명령어를 실행하고, 이후 상기 가상 주소에 대한 페이지 접근 권한을 불가로 재설정하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 8

제5항에 있어서,

상기 가상 주소가 상기 일반 메모리 영역에 해당하면, 상기 커널에 의해 인터럽트를 처리하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 9

제1항에 있어서,

상기 할당 해제하는 단계는,

프로세스에 대한 메모리 할당 해제 요청을 수신하면, 상기 관리 대상 프로세스에 해당여부를 판단하고,

상기 관리 대상 프로세스에 해당하면, 상기 커널의 메모리 주소 공간에 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하고, 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역을 할당 해제하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 10

제9항에 있어서,

상기 메모리 오류를 검출하는 단계는,

상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역의 가상 주소에 대해서 접근 시도하는 상황이면, 제2 상세 오류 유형 플래그를 갖는 오류 정보를 출력하는 것을 특징으로 하는 커널 메모리의 실시간 오류 검출 방법.

#### 청구항 11

커널 메모리의 오류 검출 장치에 있어서,

관리 대상 프로세스를 등록하는 관리 대상 등록 모듈;

상기 관리 대상 프로세스를 기준으로 상기 커널의 메모리 주소 공간에 메모리 영역을 할당하는 메모리 할당 모듈;

상기 커널의 인터럽트 핸들러를 이용하여 상기 메모리 영역에 대해서 메모리 오류를 검출하는 메모리 오류 검출 모듈; 및

상기 메모리 영역에 대해서 할당 해제하는 메모리 해제 모듈을 포함하며,

상기 메모리 할당 모듈은,

실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하지 않으면 상기 커널의 메모리 주소 공간의 일반 메모리 영역에 메모리를 할당하고,

상기 실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하면 상기 커널의 메모리 주소 공간의 검출 대상 메모리 영역에 메모리를 할당하는 것을 특징으로 하는 커널 메모리의 오류 검출 장치.

## 청구항 12

제11항에 있어서,

복수의 프로세스가 동시에 실행되는 상황에서, 상기 복수의 프로세스가 상기 커널의 메모리 주소 공간을 공유하는 것을 특징으로 하는 커널 메모리의 오류 검출 장치.

## 청구항 13

삭제

## 청구항 14

제11항에 있어서,

상기 메모리 할당 모듈은 상기 검출 대상 메모리 영역에 레드존을 설정하고, 상기 커널의 메모리 주소 공간에 상기 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하고, 상기 커널의 메모리 주소 공간의 가상 주소에 대한 페이지 접근 권한을 변경하고,

상기 메모리 오류 검출 모듈은 상기 커널의 인터럽트 핸들러에 의한 메모리 오류 검사를 통해 접근을 시도하는 가상 주소가 상기 검출 대상 메모리 영역인지 상기 일반 메모리 영역인지 판별하는 것을 특징으로 하는 커널 메모리의 오류 검출 장치.

## 청구항 15

제11항에 있어서,

상기 메모리 해제 모듈은,

프로세스에 대한 메모리 할당 해제 요청을 수신하면, 상기 관리 대상 프로세스에 해당여부를 판단하고,

상기 관리 대상 프로세스에 해당하면, 상기 커널의 메모리 주소 공간에 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하고, 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역을 할당 해제하는 것을 특징으로 하는 커널 메모리의 오류 검출 장치.

## 발명의 설명

### 기술 분야

[0001] 본 발명이 속하는 기술 분야는 운영체제 커널 메모리의 실시간 오류 검출 방법에 관한 것이다.

### 배경 기술

[0002] 이 부분에 기술된 내용은 단순히 본 실시예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0003] 소프트웨어의 버그를 찾기 위해 퍼징(Fuzzing)과 콘콜릭 실행(Concolic Execution)이 널리 사용되고 있다. 퍼징과 콘콜릭 실행을 사용해 새로운 테스트 입력을 생성하고 만약 생성된 테스트 입력을 사용하여 프로그램을 실행시켰을 때 프로그램이 비정상 동작을 일으킨다면 소프트웨어 내의 버그가 있다고 판단한다. 이때 프로그램의 비정상 동작은 프로그램이 강제로 종료되는 것을 말한다. 이 같은 동적 테스트 방법은 버그를 발견하기 위해서 프로그램의 비정상 동작 탐지가 필요하므로 프로그램의 비정상 동작이 발생하지 않는 버그는 발견하지 못한다는 문제가 존재한다.

[0004] 제시된 문제를 해결하기 위해 프로그램에서 강제 종료가 발생하지 않는 메모리 오류 버그를 발견하기 위해 다양한 메모리 오류 탐지 방법들이 제안되었다. 구글에서는 버퍼 오버플로우와 같은 메모리 오류 발생을 검출하기 위한 어드레스 새니타이저(Address Sanitizer)를 발표하였다.

[0005] 어드레스 새니타이저는 유저 영역에서 실행되는 하나의 프로세스를 대상으로 메모리 오류 감지를 수행한다. 메

모리 오류 감지를 수행하기 위해 유저 프로그램을 컴파일할 때 메모리 할당과 해제에 관련된 함수를 변경한다. 변경된 메모리 할당 함수는 메모리 할당시 할당되는 메모리 공간 앞뒤로 레드존을 생성하고 메모리 주소 영역에 대한 새도우 메모리 영역을 생성한다.

[0006] 새도우 메모리는 메모리의 접근 가능 여부에 대한 정보를 저장하고 있다. 이후 메모리 읽기/쓰기 명령어가 실행 되면 접근이 발생한 메모리 주소에 대한 새도우 메모리 값을 검사한 뒤 접근이 가능한 주소면 정상적으로 프로그램을 실행하고 접근이 불가능한 주소면 오류 메시지를 출력한 뒤 프로그램을 종료시킨다.

[0007] 어드레스 새니타이저는 소스 코드가 있는 소프트웨어에만 적용이 가능하고 소스 코드가 공개되지 않은 운영체제에 적용하는 것은 곤란한 실정이다.

## 선행기술문헌

### 특허문헌

[0008] (특허문헌 0001) 한국등록특허공보 제10-2044075호 (2019.11.06.)

## 발명의 내용

### 해결하려는 과제

[0009] 본 발명의 실시예들은 소스 코드 없이 실시간으로 대상 프로세스만을 선택하고 대상 프로세스에 대해서만 운영체제 커널의 메모리 오류를 검사하여 커널의 성능 영향을 최소화하는 데 주된 목적이 있다.

[0010] 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범위 내에서 추가적으로 고려될 수 있다.

### 과제의 해결 수단

[0011] 본 실시예의 일 측면에 의하면, 커널 메모리의 실시간 오류 검출 방법에 있어서, 관리 대상 프로세스를 등록하는 단계; 상기 관리 대상 프로세스를 기준으로 상기 커널의 메모리 주소 공간에 메모리 영역을 할당하는 단계; 상기 커널의 인터럽트 핸들러를 이용하여 상기 메모리 영역에 대해서 메모리 오류를 검출하는 단계; 및 상기 메모리 영역에 대해서 할당 해제하는 단계를 포함하는 커널 메모리의 실시간 오류 검출 방법을 제공한다.

[0012] 복수의 프로세스가 동시에 실행되는 상황에서, 상기 복수의 프로세스가 상기 커널의 메모리 주소 공간을 공유할 수 있다.

[0013] 상기 메모리 영역을 할당하는 단계는, 실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하지 않으면 상기 커널의 메모리 주소 공간에 일반 메모리 영역을 할당하는 단계; 및 상기 실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하면 상기 커널의 메모리 주소 공간에 검출 대상 메모리 영역을 할당하는 단계를 포함할 수 있다.

[0014] 상기 메모리 영역을 할당하는 단계는, 상기 검출 대상 메모리 영역에 레드존을 설정하는 단계; 및 상기 커널의 메모리 주소 공간에 상기 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하는 단계를 포함할 수 있다.

[0015] 상기 메모리 영역을 할당하는 단계는, 상기 커널의 메모리 주소 공간의 가상 주소에 대한 페이지 접근 권한을 변경하는 단계를 포함하고, 상기 메모리 오류를 검출하는 단계는, 상기 커널의 인터럽트 핸들러에 의한 메모리 오류 검사를 통해 접근을 시도하는 가상 주소가 상기 검출 대상 메모리 영역인지 상기 일반 메모리 영역인지 판별하는 단계를 포함할 수 있다.

[0016] 상기 메모리 오류를 검출하는 단계는, 상기 가상 주소가 상기 검출 대상 메모리 영역에 해당하면, 상기 새도우 메모리 영역을 검사하여 메모리 오류를 확인하고, 오류에 해당하면 제1 상세 오류 유형 플래그를 갖는 오류 정보를 출력할 수 있다.

[0017] 상기 가상 주소가 상기 검출 대상 메모리 영역에 해당하면, 상기 새도우 메모리 영역을 검사하여 메모리 오류를 확인하고, 오류에 해당하지 않으면 상기 가상 주소에 대한 페이지 접근 권한을 가능으로 변경한 후, 단일 명령

어를 실행하고, 이후 상기 가상 주소에 대한 페이지 접근 권한을 불가로 재설정할 수 있다.

[0018] 상기 가상 주소가 상기 일반 메모리 영역에 해당하면, 상기 커널에 의해 인터럽트를 처리할 수 있다.

[0019] 상기 할당 해제하는 단계는, 프로세스에 대한 메모리 할당 해제 요청을 수신하면, 상기 관리 대상 프로세스에 해당여부를 판단하고, 상기 관리 대상 프로세스에 해당하면, 상기 커널의 메모리 주소 공간에 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하고, 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역을 할당 해제할 수 있다.

[0020] 상기 메모리 오류를 검출하는 단계는, 상기 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역의 가상 주소에 대해서 접근 시도하는 상황이면, 제2 상세 오류 유형 플래그를 갖는 오류 정보를 출력할 수 있다.

[0021] 본 실시예의 다른 측면에 의하면, 커널 메모리의 오류 검출 장치에 있어서, 관리 대상 프로세스를 등록하는 관리 대상 등록 모듈; 상기 관리 대상 프로세스를 기준으로 상기 커널의 메모리 주소 공간에 메모리 영역을 할당하는 메모리 할당 모듈; 상기 커널의 인터럽트 핸들러를 이용하여 상기 메모리 영역에 대해서 메모리 오류를 검출하는 메모리 오류 검출 모듈; 및 상기 메모리 영역에 대해서 할당 해제하는 메모리 해제 모듈을 포함하는 커널 메모리의 오류 검출 장치를 제공한다.

### 발명의 효과

[0022] 이상에서 설명한 바와 같이 본 발명의 실시예들에 의하면, 소스 코드 없이 실시간으로 대상 프로세스만을 선택하고 대상 프로세스에 대해서만 운영체제 커널의 메모리 오류를 검사하여 커널의 성능 영향을 최소화할 수 있는 효과가 있다.

[0023] 여기에서 명시적으로 언급되지 않은 효과라 하더라도, 본 발명의 기술적 특징에 의해 기대되는 이하의 명세서에서 기재된 효과 및 그 잠정적인 효과는 본 발명의 명세서에 기재된 것과 같이 취급된다.

### 도면의 간단한 설명

[0024] 도 1은 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치를 예시한 블록도이다.

도 2는 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 관리 대상 등록 모듈을 예시한 도면이다.

도 3 및 도 4는 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 메모리 할당 모듈을 예시한 도면이다.

도 5는 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치가 처리하는 메모리 영역을 예시한 도면이다.

도 6은 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 메모리 오류 검출 모듈을 예시한 도면이다.

도 7은 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 메모리 해제 모듈을 예시한 도면이다.

도 8 내지 도 10은 본 발명의 다른 실시예에 따른 커널 메모리의 실시간 오류 검출 방법을 예시한 흐름도이다.

### 발명을 실시하기 위한 구체적인 내용

[0025] 이하, 본 발명을 설명함에 있어서 관련된 공지기능에 대하여 이 분야의 기술자에게 자명한 사항으로서 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하고, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다.

[0026] 기존 메모리 오류 검출 기술은 컴파일 과정에서 메모리 검사 코드를 추가하므로 버그 발견 대상의 소스 코드가 필요하다. 운영체제는 소스 코드가 공개되지 않으므로, 내부 관련 개발자가 아니면 컴파일을 통해 소스 코드를 적용하는 것이 곤란하다.

[0027] 본 실시예는 메모리 오류 검사에 소스 코드가 필요하지 않으므로, 운영체제에 적용할 수 있다. 운영체제의 소스 코드가 공개되었더라도 소스 코드가 제공되지 않은 장치 드라이버와 같은 제3자가 제공하는 소프트웨어 테스트에도 적용 가능하다.

[0028] 운영체제 커널은 여러 개의 프로세스가 동시에 실행되므로 기존 메모리 오류 검출 기술은 적용하게 되면 큰 성

능 저하가 발생한다. 커널은 프로세스 관리, 메모리 관리, 장치 관리 등의 여러 개의 기능을 동시에 수행하고 있으며 이때 모든 기능이 한 번에 테스트에 사용되지 않는다. 따라서 테스트 대상 프로세스에 한정되지 않고 커널 프로세스 전체를 검사 대상으로 포함하는 것은 주소 공간의 낭비는 물론 속도 측면에서 매우 큰 손해를 보게 된다.

- [0029] 커널은 해당 운영체제 상에서 이루어지는 모든 작업에 관여하기 때문에 사용자 프로세스(user process)와 달리 시간-민감(time-critical)한 문제는 물론 다른 모든 작업에 있어 전반적인 지연을 초래한다. 이러한 지연은 곧 퍼져와 같은 오류 검출 도구로 메모리 오류를 검출하는데 더 많은 시간을 요구하게 된다. 즉, 기존 기술은 커널 전체에 대해서 메모리 오류 검사를 수행하므로 테스트 대상이 아닌 기능에 대한 메모리 오류 검사를 수행하고 이는 성능에 직결된다.
- [0030] 본 실시예는 커널의 모든 프로세스에 대해 메모리 오류 검사를 수행하지 않고 특정 프로세스를 선택하여 메모리 오류 검사를 수행하여 커널의 성능 저하를 최소화한다.
- [0031] 본 실시예는 소프트웨어의 비정상 동작이 발생하는 버그 중 사용자가 버그를 발견하고자 하는 테스트 대상 커널 기능 또는 모듈에 대해서만 선택적으로 메모리 오류 검출을 수행하는 방식을 통해 운영체제 커널에 존재하는 버그 중 소프트웨어의 비정상 동작이 발생하지 않는 버그를 발견한다. 발견 가능한 버그의 유형으로는 힙 버퍼 오버플로우(heap buffer overflow), 해제 후 사용(use-after-free), 임계 구역을 초과한 읽기/쓰기(out-of-bounds read/write)가 있다. 출력되는 오류 정보의 상세 오류 유형 플래그를 이용하여 구분이 가능하다.
- [0032] 도 1은 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치를 예시한 블록도이다.
- [0033] 커널 메모리의 오류 검출 장치(100)는 운영체제 커널에서 사용되는 동적 메모리를 관리한다. 커널 메모리의 오류 검출 장치(100)는 관리 대상 프로세스를 등록하는 관리 대상 등록 모듈(110), 관리 대상 프로세스를 기준으로 커널의 메모리 주소 공간에 메모리 영역을 할당하는 메모리 할당 모듈(120), 커널의 인터럽트 핸들러를 이용하여 메모리 영역에 대해서 메모리 오류를 검출하는 메모리 오류 검출 모듈(130), 및 메모리 영역에 대해서 할당 해제하는 메모리 해제 모듈(140)을 포함한다.
- [0034] 관리 대상 등록 모듈(110)은 어떤 프로세스(실행파일)가 메모리 검사 대상이 될지 정한다.
- [0035] 메모리 할당 모듈(120)은 메모리 할당 단계에서는 메모리 할당시 메모리 오류 검출을 위한 추가적인 작업을 수행한다. 이때 메모리 할당시 추가적인 작업을 수행하기 위하여 커널 메모리 할당 함수들에 대해 후킹, 인라인 패치 등의 방법을 사용할 수 있다.
- [0036] 메모리 오류 검출 모듈(130)은 메모리 영역이 사용될 때 메모리 오류가 발생하는지를 검사한다.
- [0037] 메모리 해제 모듈(140)은 메모리 검출을 위해 사용된 추가적인 자원들을 정리한다. 이때 커널 메모리 해제 함수들에 대해 후킹, 인라인 패치 등의 방법을 사용할 수 있다.
- [0038] 도 2는 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 관리 대상 등록 모듈을 예시한 도면이다.
- [0039] 관리 대상 등록 모듈(110)은 외부 요청(퍼져, 커널 모듈 개발자 등)을 통해 메모리 오류 검사를 지정할 대상을 관리 모듈이 전달받는다. 이때 요청은 이미 실행되고 있는 커널 프로세스, 실행 파일 형태의 커널 모듈이 될 수 있다. 이후 검출 장치에 pid 등 프로세스 간 식별 가능한 정보를 등록하여 메모리 관리 대상 등록을 완료한다.
- [0040] 도 3 및 도 4는 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 메모리 할당 모듈을 예시한 도면이다.
- [0041] 메모리 할당 모듈(120)은 커널에서 메모리 할당이 요청되면 관리 대상에 등록된 프로세스의 요청인지 확인한다. 만약 등록된 프로세스의 요청이라면 메모리 할당 함수가 호출될 때 본 실시예에 따라 정의된 알고리즘에 따른 함수가 메모리 할당 함수가 실행되기 전/후 호출된다. 이를 통해 메모리 오류 검사를 선택적으로 수행하여 더욱 효율적인 메모리 오류 검출을 가능하게 한다.
- [0042] 프로세스 필터링 모듈은 커널에서 메모리 할당 관련 함수가 호출되면 인라인 후킹을 통해 해당 메모리 영역이 테스트할 프로세스의 영역인지 확인한다. 테스트 대상 프로세스로 확인되면 대상 메모리 할당 모듈에 요청받은 할당 크기를 전송하며 그렇지 않다면 일반적인 메모리 할당을 수행한다.
- [0043] 도 5는 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치가 처리하는 메모리 영역을 예시한 도면이다.
- [0044] 프로세스 필터링 결과 해당 프로세스가 테스트 프로세스라면 메모리 할당 함수가 실행된 후 메모리 오류를 검출



하기 위한 추가적인 동작이 수행된다. 먼저 메모리 할당을 수행할 때 요청된 메모리의 크기보다 더 큰 크기의 메모리를 기존 메모리 영역과 분리된 영역에 할당한다.

- [0045] 커널 메모리 영역과 메모리 생성 모듈이 해당 영역에 할당하는 과정을 살펴보면, 커널 어드레스 새니타이저는 할당 영역을 구분하지 않고 모든 메모리 할당을 일반적인 메모리 할당 영역에서 진행하게 된다. 이와 다르게 본 실시예는 검출 대상 프로세스가 사용하는 오류 검출 대상 메모리 영역을 구분하여 사용한다.
- [0046] 도 5에서 오류 검출 대상 메모리 영역이 일반적으로 할당되는 메모리와 구분된 영역에 할당되는 것을 확인할 수 있다. 이러한 작업을 통해 검사 대상 프로세스에 속하는 객체만 메모리 오류 검출을 수행한다. 검출 대상 프로세스가 할당하는 객체에 대해서만 레드존을 생성한다.
- [0047] 레드존을 효율적으로 검사하기 위해 새도우 메모리는 기존 방법과 다르게 오류 검출 대상 메모리 할당 영역에 대해서만 생성되고 해당 영역 메모리의 접근 가능여부를 기록하고 있다. 레드존과 새도우 메모리를 생성한 뒤 마지막으로 할당된 메모리의 권한을 변경한다. 메모리의 권한을 접근 불가로 변경하여 해당 메모리에 접근이 발생하면 인터럽트를 발생시켜 메모리 오류 검출을 수행한다.
- [0048] 도 6은 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 메모리 오류 검출 모듈을 예시한 도면이다.
- [0049] 대상 메모리 할당 모듈이 메모리 오류를 검출하고자 하는 메모리 영역에 대해서 접근 권한을 제거했으므로 프로세스에서 해당 메모리에 읽기 또는 쓰기와 같은 접근을 시도하면 페이지 접근 불가 인터럽트가 발생하게 된다.
- [0050] 인터럽트가 발생할 경우 페이지 오류 핸들러는 주소 검사 장치로 해당 메모리 오류 정보를 전달한다. 이 때 전달되는 정보는 접근 시도한 메모리의 주소이며 해당 정보를 이용하여 주소 검사 장치는 관리 대상 메모리에서 발생한 인터럽트인지 확인한다.
- [0051] 만약 관리 대상 메모리에서 발생했다면 해당 주소에 대한 새도우 메모리 검사를 수행하여 메모리 오류가 발생하였는지 확인하고 오류가 발생하였다면 오류 정보를 출력한다. 이때 오류 정보는 오류 발생 주소, 상세 오류 종류, 새도우 메모리 정보가 포함한다.
- [0052] 만약 관리 대상 메모리에서 발생한 인터럽트가 아닌 경우, 커널의 원래 페이지 접근 불가 오류 핸들러를 호출하며 이후 처리는 커널에 맡긴다.
- [0053] 새도우 메모리 확인 과정을 마친 후 임시로 페이지를 접근 가능하게 만들며 이후 단일 실행을 통해 명령을 실행한다. 그 후 다시 페이지를 접근 불가로 설정한 후 종료한다. 이를 통해 같은 가상 주소에 대한 반복적인 접근에 대해서도 메모리 오류 검출을 수행할 수 있다.
- [0054] 도 7은 본 발명의 일 실시예에 따른 커널 메모리의 오류 검출 장치의 메모리 해제 모듈을 예시한 도면이다.
- [0055] 메모리의 사용이 끝나고 관리 대상 등록 모듈(110)로 메모리 할당 해제가 요청되면 테스트 대상 프로세스인지를 필터링한다. 필터링한 뒤 테스트 대상 프로세스일 경우 메모리 오류 검출을 위한 메모리 할당이 이루어졌기 때문에 이를 처리하기 위한 메모리 해제 모듈에서 새도우 메모리 설정과 요청 메모리 할당 해제 과정을 거치며 테스트 대상 프로세스가 아닐 경우 일반적인 메모리 할당 해제를 수행한다.
- [0056] 도 8 내지 도 10은 본 발명의 다른 실시예에 따른 커널 메모리의 실시간 오류 검출 방법을 예시한 흐름도이다. 커널 메모리의 실시간 오류 검출 방법은 커널 메모리의 오류 검출 장치에 의해 수행될 수 있다.
- [0057] 도 8을 참조하면, 단계 S210에서 관리 대상 프로세스를 등록한다. 단계 S220에서 관리 대상 프로세스를 기준으로 커널의 메모리 주소 공간에 메모리 영역을 할당한다. 단계 S230에서 커널의 인터럽트 핸들러를 이용하여 메모리 영역에 대해서 메모리 오류를 검출한다. 단계 S240에서 메모리 영역에 대해서 할당 해제한다.
- [0058] 도 9를 참조하면, 단계 S310에서 관리 대상 프로세스를 등록한다. 복수의 프로세스가 동시에 실행되는 상황에서, 복수의 프로세스가 커널의 메모리 주소 공간을 공유한다.
- [0059] 단계 S320에서 메모리 할당 요청 프로세스를 판별한다. 메모리 영역을 할당하는 단계는, 실행 요청된 프로세스가 관리 대상 프로세스에 해당하지 않으면 커널의 메모리 주소 공간의 일반 메모리 영역에 메모리를 할당하는 단계를 포함하고, 실행 요청된 프로세스가 상기 관리 대상 프로세스에 해당하면 커널의 메모리 주소 공간의 검출 대상 메모리 영역에 메모리를 할당하는 단계를 포함한다.
- [0060] 단계 S330에서 레드존 및 새도우 메모리를 생성한다. 메모리 영역을 할당하는 단계는 검출 대상 메모리 영역에

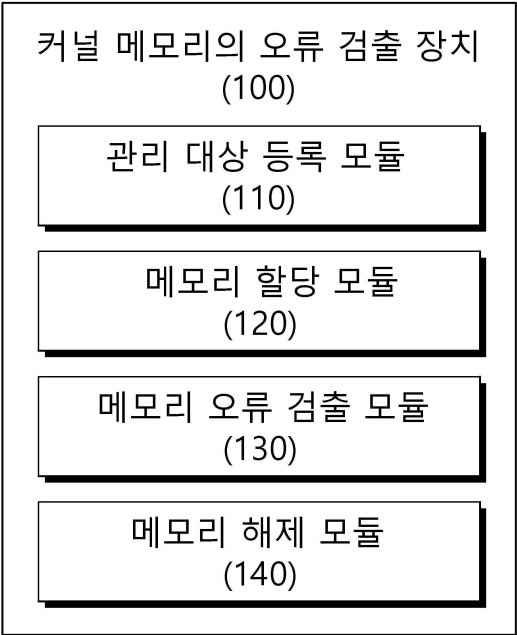
레드존을 설정하는 단계를 포함하고, 커널의 메모리 주소 공간에 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하는 단계를 포함한다.

- [0061] 단계 S340에서 할당 요청된 메모리의 가상 주소에 대한 페이지 접근 불가를 설정한다. 메모리 영역을 할당하는 단계는 커널의 메모리 주소 공간의 가상 주소에 대한 페이지 접근 권한을 변경하는 단계를 포함한다.
- [0062] 메모리 오류를 검출하는 단계는 커널의 인터럽트 핸들러에 의한 메모리 오류 검사를 통해 접근을 시도하는 가상 주소가 검출 대상 메모리 영역인지 일반 메모리 영역인지 판별하는 단계를 포함한다.
- [0063] 도 10을 참조하면, 단계 S410에서 페이지 접근 불가 오류가 발생했는지 판단한다. 단계 S420에서 모든 페이지 접근 불가 오류에 대한 관리 대상 메모리를 판별한다. 단계 S430에서 관리 대상 메모리의 새도우 메모리 정보를 통한 메모리 오류를 감지한다. 단계 S440에서 메모리 오류 정보를 출력한다.
- [0064] 메모리 오류를 검출하는 단계는, 가상 주소가 상기 검출 대상 메모리 영역에 해당하면, 새도우 메모리 영역을 검사하여 메모리 오류를 확인하고, 오류에 해당하면 제1 상세 오류 유형 플래그(임계 구역 초과)를 갖는 오류 정보를 출력한다.
- [0065] 단계 S450에서 페이지 접근 불가 오류가 발생한 가상 주소에 대한 페이지 접근 가능을 설정한다. 단계 S460에서 단일 명령어를 실행한다. 단계 S470에서 페이지 접근 불가 오류가 발생한 가상 주소에 대한 페이지 접근 불가를 설정한다.
- [0066] 가상 주소가 검출 대상 메모리 영역에 해당하면, 새도우 메모리 영역을 검사하여 메모리 오류를 확인하고, 오류에 해당하지 않으면 가상 주소에 대한 페이지 접근 권한을 가능으로 설정하고, 단일 명령어를 실행하고, 이후 가상 주소에 대한 페이지 접근 권한을 불가로 재설정한다.
- [0067] 가상 주소가 일반 메모리 영역에 해당하면, 커널에 의해 인터럽트를 처리한다.
- [0068] 단계 S510에서 메모리 할당 해제 요청을 판단한다. 단계 S520에서 모든 메모리 할당 해제 요청에 대한 관리 대상 메모리를 판별한다. 단계 S530에서 해제 요청된 메모리 주소에 대한 새도우 메모리를 설정한다. 단계 S540에서 해제 요청된 메모리 주소에 대한 메모리를 해제한다.
- [0069] 할당 해제하는 단계는, 프로세스에 대한 메모리 할당 해제 요청을 수신하면, 실행 요청된 프로세스가 관리 대상 프로세스인지 해당여부를 판단하고, 관리 대상 프로세스에 해당하면, 커널의 메모리 주소 공간에 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역을 할당하고, 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역을 할당 해제한다.
- [0070] 메모리 오류를 검출하는 단계는, 메모리 할당 해제 요청에 대응하는 검출 대상 메모리 영역에 대한 새도우 메모리 영역의 가상 주소에 대해서 접근 시도하는 상황이면, 제2 상세 오류 유형 플래그(해제 후 사용)를 갖는 오류 정보를 출력한다.
- [0071] 커널 메모리의 오류 검출 장치는 적어도 하나의 프로세서, 컴퓨터 판독 가능한 저장매체 및 통신 버스를 포함할 수 있다.
- [0072] 프로세서는 커널 메모리의 오류 검출 장치로 동작하도록 제어할 수 있다. 예컨대, 프로세서는 컴퓨터 판독 가능한 저장 매체에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 컴퓨터 실행 가능 명령어는 프로세서에 의해 실행되는 경우 커널 메모리의 오류 검출 장치로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.
- [0073] 컴퓨터 판독 가능한 저장 매체는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능한 저장 매체에 저장된 프로그램은 프로세서에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독한 가능 저장 매체는 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 커널 메모리의 오류 검출 장치에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.
- [0074] 통신 버스는 프로세서, 컴퓨터 판독 가능한 저장 매체를 포함하는 커널 메모리의 오류 검출 장치의 다른 다양한 컴포넌트들을 상호 연결한다.

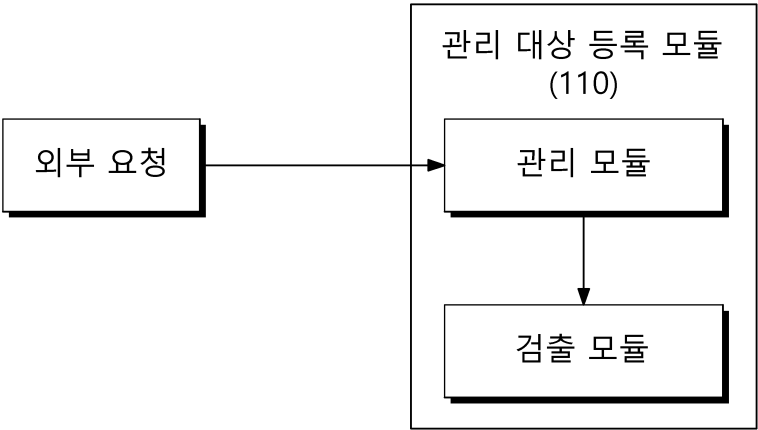
- [0075] 커널 메모리의 오류 검출 장치는 또한 하나 이상의 입출력 장치를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스 및 하나 이상의 통신 인터페이스를 포함할 수 있다. 입출력 인터페이스 및 통신 인터페이스는 통신 버스에 연결된다. 입출력 장치는 입출력 인터페이스를 통해 커널 메모리의 오류 검출 장치의 다른 컴포넌트들에 연결될 수 있다.
- [0076] 커널 메모리의 오류 검출 장치는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를 포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다.
- [0077] 커널 메모리의 오류 검출 장치는 하드웨어적 요소가 마련된 컴퓨팅 디바이스 또는 서버에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 디바이스 또는 서버는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모듈 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다.
- [0078] 도 8 내지 도 10에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 8 내지 도 10에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이다.
- [0079] 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능한 매체에 기록될 수 있다. 컴퓨터 판독 가능한 매체는 실행을 위해 프로세서에 명령어를 제공하는 데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예를 들면, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다.
- [0080] 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

도면

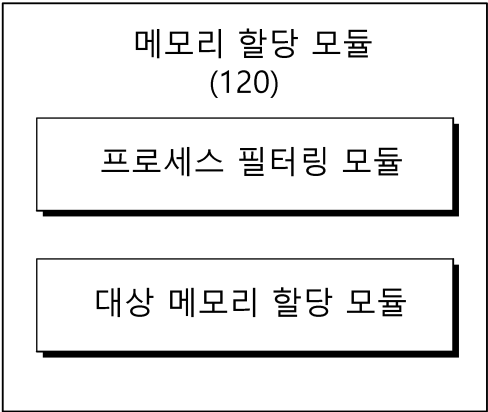
도면1



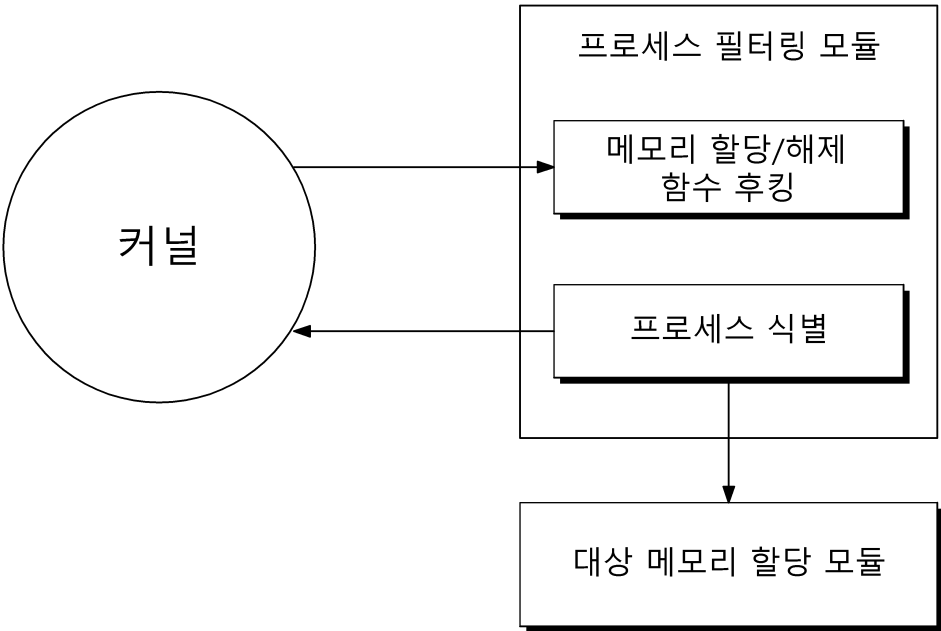
도면2



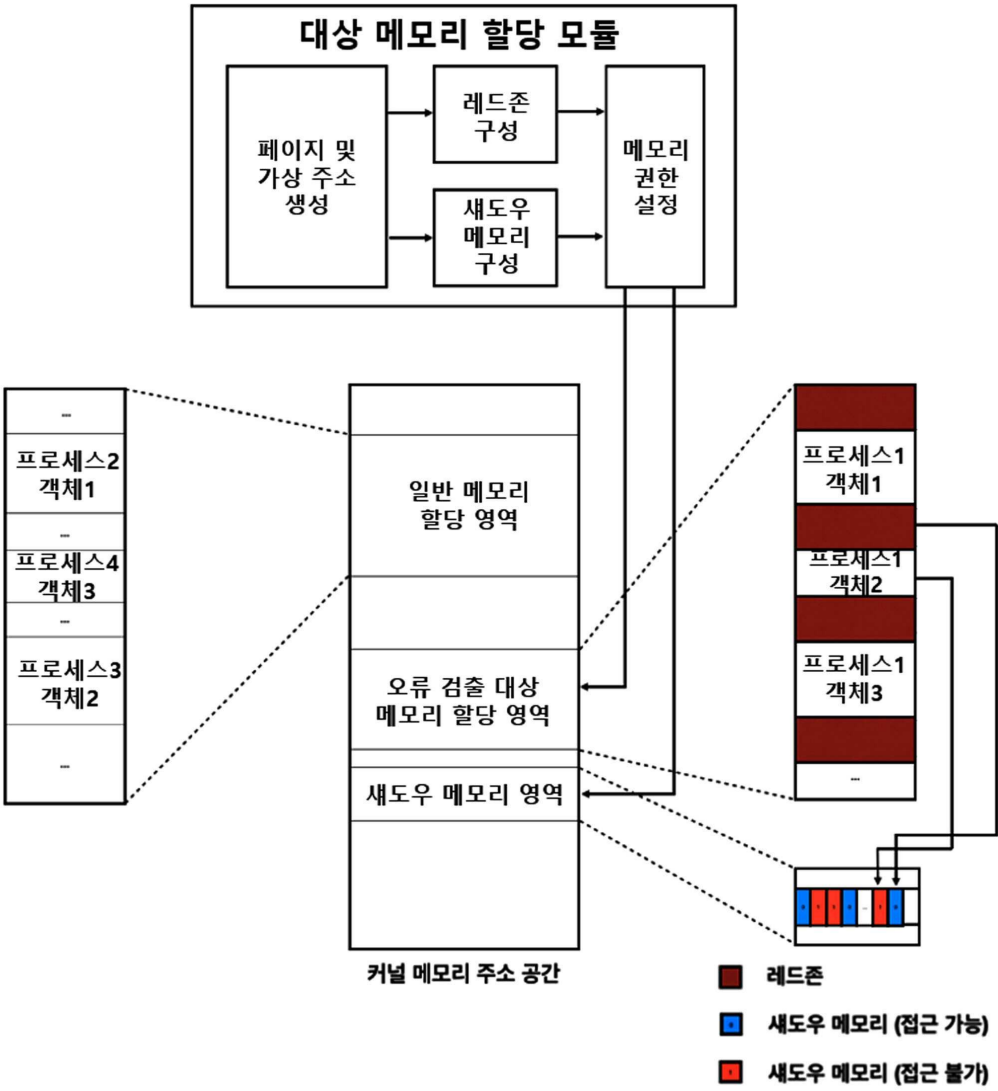
도면3



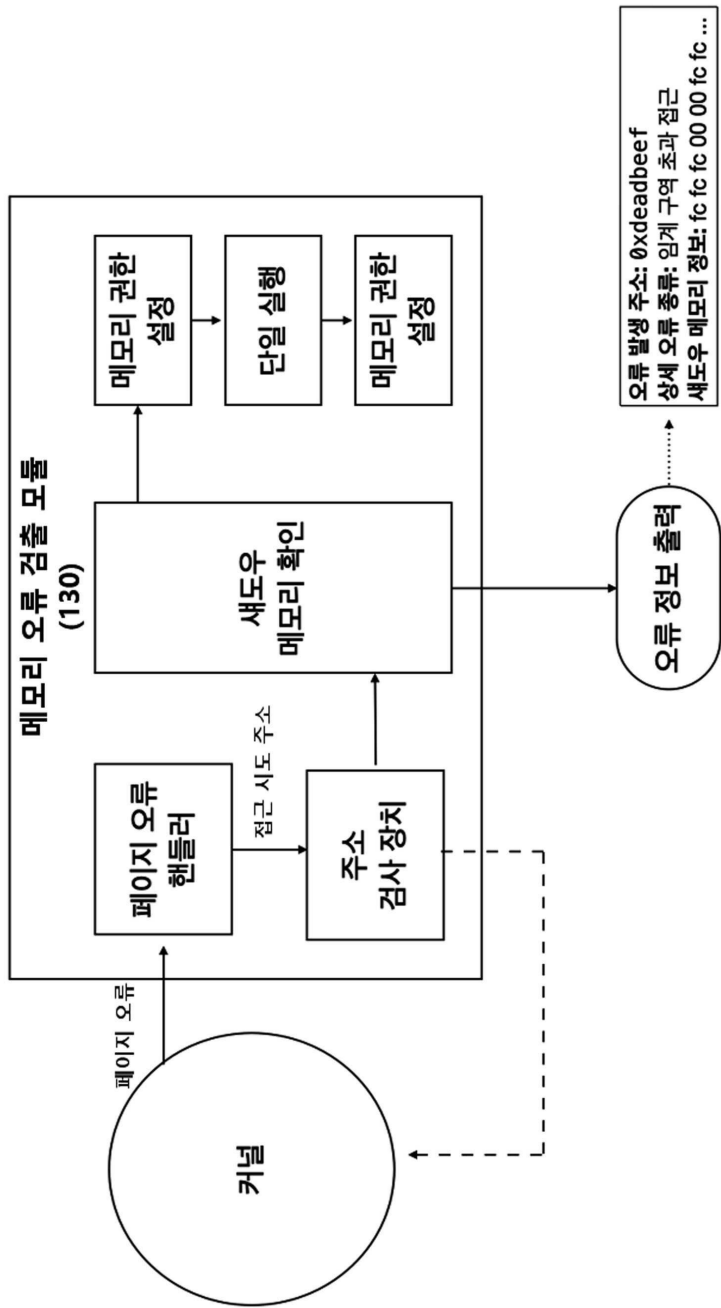
도면4



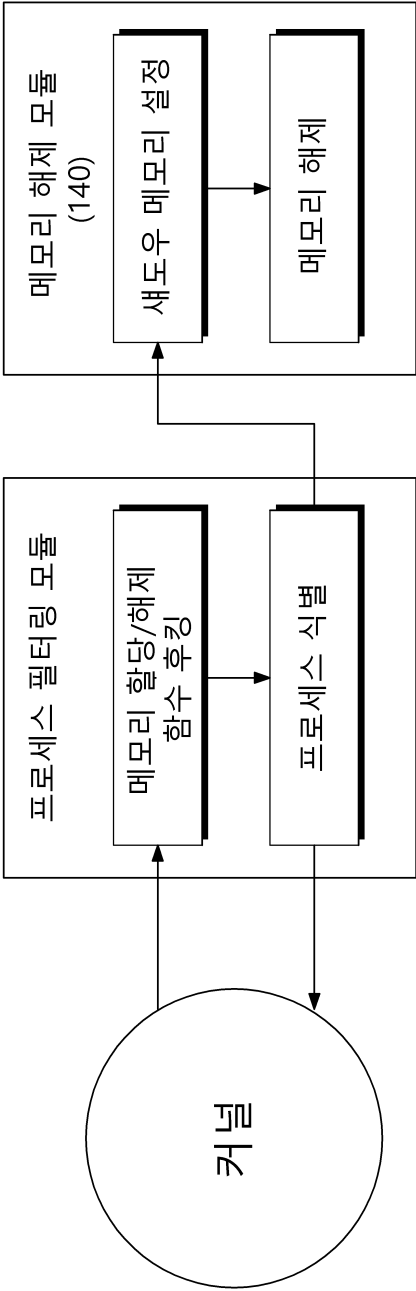
도면5



도면6

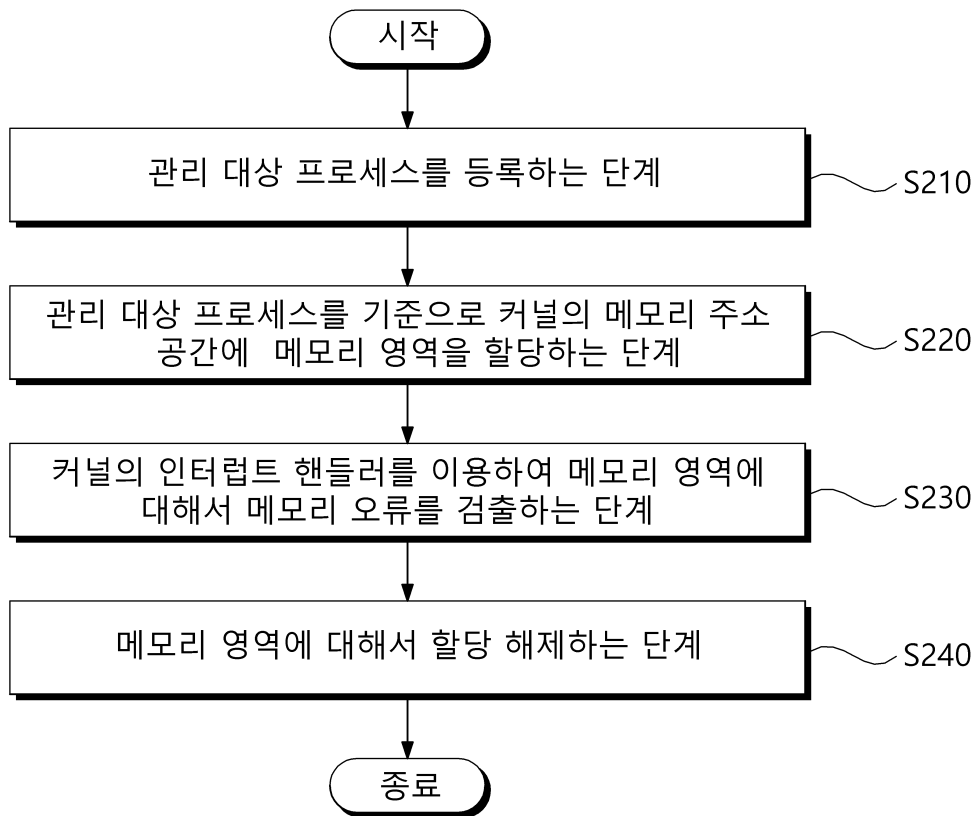


도면7

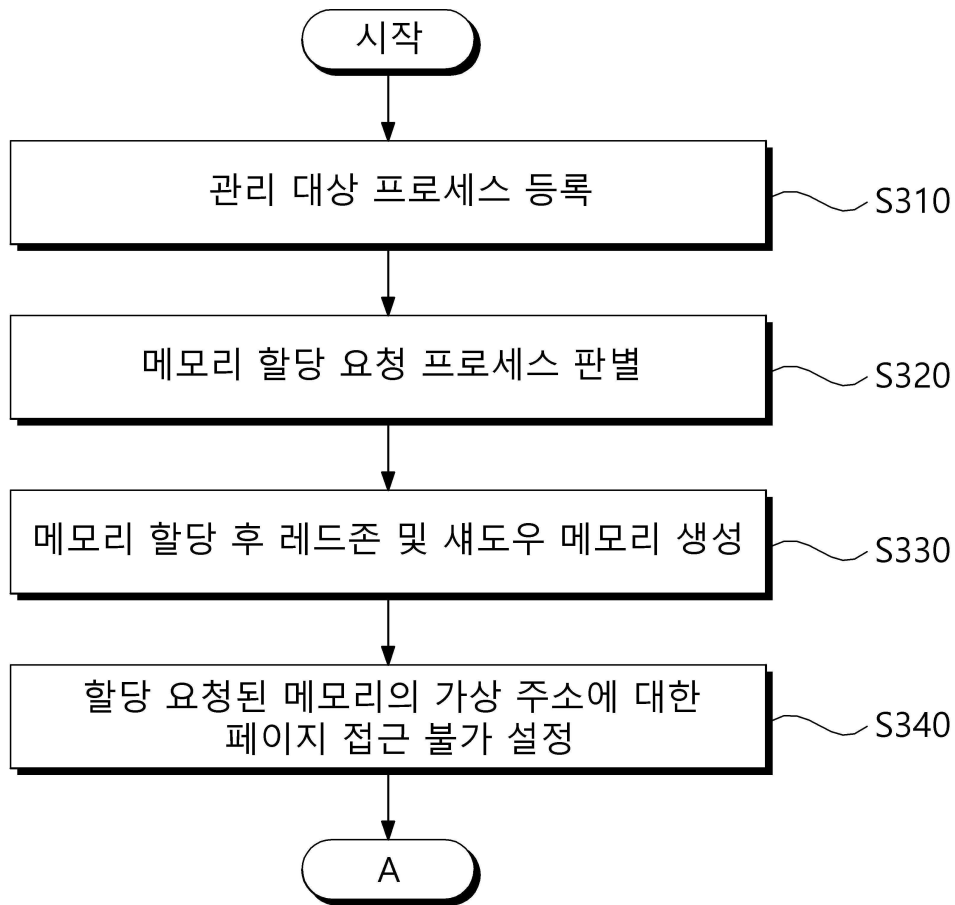




도면8



도면9



도면10

