



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0027993
(43) 공개일자 2018년03월15일

(51) 국제특허분류(Int. Cl.)
G06F 11/14 (2006.01)
(52) CPC특허분류
G06F 11/1407 (2013.01)
G06F 11/1471 (2013.01)
(21) 출원번호 10-2017-0048313
(22) 출원일자 2017년04월14일
심사청구일자 2017년04월14일
(30) 우선권주장
1020160114917 2016년09월07일 대한민국(KR)

(71) 출원인
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
노삼혁
울산광역시 울주군 언양읍 유니스트길 50
문영제
울산광역시 울주군 언양읍 유니스트길 50
(뒷면에 계속)
(74) 대리인
김태현

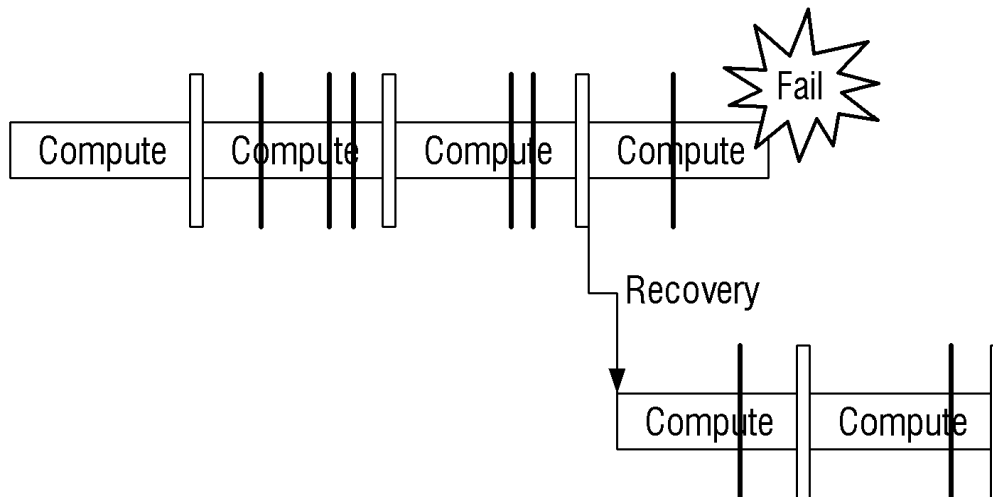
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 전자 장치 및 제어 방법

(57) 요약

전자 장치 및 제어 방법이 개시된다. 본 개시는 메인 메모리 영역 및 파일 시스템 영역을 포함하는 뉴메모리 및 연산을 수행하고, 기 설정된 조건에 따라 메인 메모리 영역 및 파일 시스템 영역에 대해 체크 포인팅을 수행하는 프로세서를 포함하며, 메인 메모리 영역은 데이터가 변경된 페이지 정보를 라이팅(writing)하는 체크 포인트 테이블을 포함하고, 프로세서는 메인 메모리 영역에 대해 체크 포인팅을 수행할 때, 체크 포인트 테이블을 초기화하고 메인 메모리 영역의 페이지의 속성(attribute)을 읽기 전용(read-only)으로 설정하는 전자 장치를 제공한다.

대표도 - 도2



(72) 발명자

박재형

울산광역시 울주군 언양읍 유니스트길 50

송현섭

울산광역시 울주군 언양읍 유니스트길 50

김정현

광주광역시 광산구 첨단중앙로181번길 42- 5 (월계동, 금호아파트)

명세서

청구범위

청구항 1

메인 메모리 영역 및 파일 시스템 영역을 포함하는 뉴메모리; 및

연산을 수행하고, 기 설정된 조건에 따라 상기 메인 메모리 영역 및 상기 파일 시스템 영역에 대해 체크 포인팅을 수행하는 프로세서;를 포함하며,

상기 메인 메모리 영역은,

데이터가 변경된 페이지 정보를 라이팅(writing)하는 체크 포인트 테이블을 포함하고,

상기 프로세서는,

상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때, 상기 체크 포인트 테이블을 초기화하고 상기 메인 메모리 영역의 페이지의 속성(attribute)을 읽기 전용(read-only)으로 설정하는, 전자 장치.

청구항 2

제1항에 있어서,

상기 프로세서는,

상기 연산을 수행할 때, 제1 페이지의 데이터 업데이트가 요청되는 경우 상기 제1 페이지의 정보를 상기 체크 포인트 테이블에 라이팅하는, 전자 장치.

청구항 3

제2항에 있어서,

상기 프로세서는,

상기 제1 페이지를 복사한 제2 페이지를 생성하고, 상기 제2 페이지의 정보를 상기 체크 포인트 테이블에 라이팅하며, 상기 속성을 라이트(write)로 변경하여 상기 제1 페이지의 데이터를 업데이트하는, 전자 장치.

청구항 4

제3항에 있어서,

상기 프로세서는,

상기 연산을 수행하거나 상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때 오류가 발생하는 경우, 상기 체크 포인트 테이블에 라이팅된 정보에 기초하여 상기 제1 페이지를 롤백(roll back)하는, 전자 장치.

청구항 5

제1항에 있어서,

상기 파일 시스템 영역은 버퍼 및 파일 저장부를 포함하고,

상기 파일 저장부는,

기 저장된 데이터 블록을 포함하며,

상기 프로세서는,

상기 데이터 블록이 업데이트되는 경우, 상기 업데이트된 데이터 블록을 상기 버퍼에 저장하는, 전자 장치.

청구항 6

제5항에 있어서,

상기 프로세서는,

상기 파일 시스템 영역에 대해 체크 포인팅을 수행할 때, 상기 업데이트된 데이터 블록을 상기 파일 저장부에 라이팅하는, 전자 장치.

청구항 7

제1항에 있어서,

상기 프로세서는,

상기 메인 메모리 영역에 대한 체크 포인팅을 수행한 후 상기 파일 시스템 영역에 대한 체크 포인팅을 수행하는, 전자 장치.

청구항 8

제1항에 있어서,

상기 기 설정된 조건은,

기 설정된 주기, 상기 데이터의 변경이 기 설정된 횟수를 만족하는 조건 중 적어도 하나인, 전자 장치.

청구항 9

체크 포인팅을 수행하는 단계; 및

연산을 수행하는 단계;를 포함하고,

상기 체크 포인팅을 수행하는 단계는,

데이터가 변경된 페이지 정보를 라이팅하는 체크 포인트 테이블을 포함하는 뉴메모리의 메인 메모리 영역에 대해 기 설정된 조건에 따라 체크 포인팅을 수행하고, 상기 뉴메모리의 파일 시스템 영역에 대해 체크 포인팅을 수행하며, 상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때, 상기 체크 포인트 테이블을 초기화하고 상기 메인 메모리 영역의 페이지의 속성(attribute)을 읽기 전용(read-only)으로 설정하는, 전자 장치의 제어 방법.

청구항 10

제9항에 있어서,

상기 연산을 수행하는 단계는,

제1 페이지의 데이터 업데이트가 요청되는 경우 상기 제1 페이지의 정보를 상기 체크 포인트 테이블에 라이팅하는, 전자 장치의 제어 방법.

청구항 11

제10항에 있어서,

상기 연산을 수행하는 단계는,

상기 제1 페이지를 복사한 제2 페이지를 생성하고, 상기 제2 페이지의 정보를 상기 체크 포인트 테이블에 라이팅하며, 상기 속성을 라이트(write)로 변경하여 상기 제1 페이지의 데이터를 업데이트하는, 전자 장치의 제어 방법.

청구항 12

제11항에 있어서,

상기 연산을 수행하는 단계는,

상기 연산을 수행하거나 상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때 오류가 발생하는 경우, 상기 체크 포인트 테이블에 라이팅된 정보에 기초하여 상기 제1 페이지를 롤백(roll back)하는, 전자 장치의 제어 방법.

법.

청구항 13

제9항에 있어서,

상기 체크 포인팅을 수행하는 단계는,

버퍼 및 파일 저장부를 포함하는 상기 파일 시스템 영역의 상기 파일 저장부에 기 저장된 데이터 블록이 업데이트되는 경우, 상기 업데이트된 데이터 블록을 상기 버퍼에 저장하는, 전자 장치의 제어 방법.

청구항 14

제13항에 있어서,

상기 체크 포인팅을 수행하는 단계는,

상기 파일 시스템 영역에 대해 체크 포인팅을 수행할 때, 상기 업데이트된 데이터 블록을 상기 파일 저장부에 라이팅하는, 전자 장치의 제어 방법.

청구항 15

제9항에 있어서,

상기 체크 포인팅을 수행하는 단계는,

상기 메인 메모리 영역에 대한 체크 포인팅을 수행한 후 상기 파일 시스템 영역에 대한 체크 포인팅을 수행하는, 전자 장치의 제어 방법.

발명의 설명

기술 분야

[0001] 본 개시는 전자 장치 및 제어 방법에 관한 것으로, 더욱 상세하게는 체크 포인팅을 이용하여 데이터를 관리하는 전자 장치 및 제어 방법에 관한 것이다.

배경 기술

[0002] 체크포인팅 기법은 예상치 못한 오류 이벤트가 발생하는 경우, 시스템이 반환할 수 있는 일관된 상태를 스토리지에 저장하기 위해 일반적으로 사용되는 메커니즘이다. 체크포인팅 기법은 시스템의 상태를 파일로 변환하고, 그 파일은 스토리지 장치에 저장된다.

[0003] 도 1을 참조하면, 종래의 체크 포인팅을 이용하여 데이터를 관리하는 방법이 도시되어 있다. 체크 포인트는 시스템의 상태를 파일로 변환하고, 변환된 파일은 저장 장치에 저장된다. 만일, 연산 중에 에러가 발생한 경우, 전자 장치는 가장 최근의 체크 포인팅에서 시스템을 복구하고 다시 시작해야 하므로 w 만큼 시간이 손실된다. 또한, 체크 포인트에서 시스템을 복원하는데 필요한 시간인 r 만큼 시간이 필요하다. 따라서, 종래의 전자 장치는 오류 발생 후 복구시까지 $w+r$ 만큼의 시간이 소요되었다. 일반적으로 HPC(High Performance Computing) 시스템의 경우 연속 연산 시간(또는, checkpoint interval)은 3시간 내지 4 시간 정도이고, 체크 포인팅 시간은 30분 내지 1시간 정도이다. 따라서, 종래의 전자 장치는 오류가 발생하는 경우, 많은 시간이 소요되는 단점이 있다.

[0004] 또한, 종래의 전자 장치는 세그먼트 정보, 프로세서의 휘발성 정보 및 장치가 모두 동기식으로 함께 저장되어야 하므로 데이터의 일관성 및 지속성을 보장하기 위해 복잡한 소프트웨어 관리 구조를 필요로 한다.

발명의 내용

해결하려는 과제

[0005] 본 개시는 상술한 문제점을 해결하기 위한 것으로 본 개시의 목적은 시스템 구동의 신뢰성을 확보하기 위해 효율적인 데이터의 일관성 및 지속성을 제공할 수 있는 전자 장치 및 제어 방법을 제공함에 있다.

과제의 해결 수단

- [0006] 이상과 같은 목적을 달성하기 위해 본 개시의 일 실시 예에 따르면, 메인 메모리 영역 및 파일 시스템 영역을 포함하는 뉴메모리 및 연산을 수행하고, 기 설정된 조건에 따라 상기 메인 메모리 영역 및 상기 파일 시스템 영역에 대해 체크 포인팅을 수행하는 프로세서를 포함하며, 상기 메인 메모리 영역은 데이터가 변경된 페이지 정보를 라이팅(writing)하는 체크 포인트 테이블을 포함하고, 상기 프로세서는 메인 메모리 영역에 대해 체크 포인팅을 수행할 때, 상기 체크 포인트 테이블을 초기화하고 상기 메인 메모리 영역의 페이지의 속성(attribute)을 읽기 전용(read-only)으로 설정하는 전자 장치를 제공한다.
- [0007] 그리고, 상기 프로세서는 상기 연산을 수행할 때, 제1 페이지의 데이터 업데이트가 요청되는 경우 상기 제1 페이지의 정보를 상기 체크 포인트 테이블에 라이팅할 수 있다.
- [0008] 그리고, 상기 프로세서는 상기 제1 페이지를 복사한 제2 페이지를 생성하고, 상기 제2 페이지의 정보를 상기 체크 포인트 테이블에 라이팅하며, 상기 속성을 라이트(write)로 변경하여 상기 제1 페이지의 데이터를 업데이트할 수 있다.
- [0009] 또한, 상기 프로세서는 상기 연산을 수행하거나 상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때 오류가 발생하는 경우, 상기 체크 포인트 테이블에 라이팅된 정보에 기초하여 상기 제1 페이지를 롤백(roll back)할 수 있다.
- [0010] 한편, 상기 파일 시스템 영역은 버퍼 및 파일 저장부를 포함하고, 상기 파일 저장부는 기 저장된 데이터 블록을 포함하며, 상기 프로세서는 상기 데이터 블록이 업데이트되는 경우, 상기 업데이트된 데이터 블록을 상기 버퍼에 저장할 수 있다.
- [0011] 그리고, 상기 프로세서는 상기 파일 시스템 영역에 대해 체크 포인팅을 수행할 때, 상기 업데이트된 데이터 블록을 상기 파일 저장부에 라이팅할 수 있다.
- [0012] 또한, 상기 프로세서는 상기 메인 메모리 영역에 대한 체크 포인팅을 수행한 후 상기 파일 시스템 영역에 대한 체크 포인팅을 수행할 수 있다.
- [0013] 한편, 상기 기 설정된 조건은 기 설정된 주기, 상기 데이터의 변경이 기 설정된 횟수를 만족하는 조건 중 적어도 하나일 수 있다.
- [0014] 이상과 같은 목적을 달성하기 위해 본 개시의 일 실시 예에 따르면, 체크 포인팅을 수행하는 단계 및 연산을 수행하는 단계를 포함하고, 상기 체크 포인팅을 수행하는 단계는 데이터가 변경된 페이지 정보를 라이팅하는 체크 포인트 테이블을 포함하는 뉴메모리의 메인 메모리 영역에 대해 기 설정된 조건에 따라 체크 포인팅을 수행하고, 상기 뉴메모리의 파일 시스템 영역에 대해 체크 포인팅을 수행하며, 상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때, 상기 체크 포인트 테이블을 초기화하고 상기 메인 메모리 영역의 페이지의 속성(attribute)을 읽기 전용(read-only)으로 설정하는 전자 장치의 제어 방법을 제공한다.
- [0015] 그리고, 상기 연산을 수행하는 단계는 제1 페이지의 데이터 업데이트가 요청되는 경우 상기 제1 페이지의 정보를 상기 체크 포인트 테이블에 라이팅할 수 있다.
- [0016] 또한, 상기 연산을 수행하는 단계는 상기 제1 페이지를 복사한 제2 페이지를 생성하고, 상기 제2 페이지의 정보를 상기 체크 포인트 테이블에 라이팅하며, 상기 속성을 라이트(write)로 변경하여 상기 제1 페이지의 데이터를 업데이트할 수 있다.
- [0017] 또한, 상기 연산을 수행하는 단계는 상기 연산을 수행하거나 상기 메인 메모리 영역에 대해 체크 포인팅을 수행할 때 오류가 발생하는 경우, 상기 체크 포인트 테이블에 라이팅된 정보에 기초하여 상기 제1 페이지를 롤백(roll back)할 수 있다.
- [0018] 한편, 상기 체크 포인팅을 수행하는 단계는 버퍼 및 파일 저장부를 포함하는 상기 파일 시스템 영역의 상기 파일 저장부에 기 저장된 데이터 블록이 업데이트되는 경우, 상기 업데이트된 데이터 블록을 상기 버퍼에 저장할 수 있다.
- [0019] 또한, 상기 체크 포인팅을 수행하는 단계는 상기 파일 시스템 영역에 대해 체크 포인팅을 수행할 때, 상기 업데이트된 데이터 블록을 상기 파일 저장부에 라이팅할 수 있다.
- [0020] 또한, 상기 체크 포인팅을 수행하는 단계는 상기 메인 메모리 영역에 대한 체크 포인팅을 수행한 후 상기 파일

시스템 영역에 대한 체크 포인팅을 수행할 수 있다.

발명의 효과

[0021] 이상 설명한 바와 같이, 본 개시의 다양한 실시 예에 따르면, 전자 장치 및 제어 방법은 데이터의 데이터 저장의 효율을 높이고 데이터의 일관성 및 지속성을 보장할 수 있다.

도면의 간단한 설명

[0022] 도 1은 종래의 연산 및 체크 포인팅을 수행하는 과정을 설명하는 도면이다.
 도 2는 본 개시의 일 실시 예에 따른 연산 및 체크 포인팅을 수행하는 과정을 설명하는 도면이다.
 도 3은 본 개시의 일 실시 예에 따른 전자 장치의 블록도이다.
 도 4는 본 개시의 일 실시 예에 따른 프로세서의 세부 블록도이다.
 도 5는 본 개시의 일 실시 예에 따른 전자 장치의 아키텍처를 설명하는 도면이다.
 도 6은 본 개시의 일 실시 예에 따른 전자 장치가 프로세스에 관한 정보를 저장하는 과정을 설명하는 도면이다.
 도 7은 본 개시의 일 실시 예에 따른 전자 장치가 프로세스에 관한 정보를 관리하는 구조를 설명하는 도면이다.
 도 8은 본 개시의 다른 실시 예에 따른 전자 장치가 데이터 및 메타 데이터를 저장하는 과정을 설명하는 도면이다.
 도 9는 본 개시의 일 실시 예에 따른 연산 및 체크 포인팅을 수행하는 과정을 구체적으로 설명하는 도면이다.
 도 10은 본 개시의 일 실시 예에 따른 전자 장치 제어 방법의 흐름도이다.
 도 11은 본 개시의 일 실시 예에 따른 이벤트에 따라 특정 프로세스에 관한 정보를 저장하는 전자 장치의 동작을 설명하는 흐름도이다.
 도 12는 본 개시의 일 실시 예에 따른 이벤트에 따라 특정 데이터 및 메타 데이터를 저장하는 전자 장치의 동작을 설명하는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0023] 이하에서는 첨부된 도면을 참조하여 다양한 실시 예를 보다 상세하게 설명한다. 본 명세서에 기재된 실시 예는 다양하게 변형될 수 있다. 특정한 실시 예가 도면에서 묘사되고 상세한 설명에서 자세하게 설명될 수 있다. 그러나, 첨부된 도면에 개시된 특정한 실시 예는 다양한 실시 예를 쉽게 이해하도록 하기 위한 것일 뿐이다. 따라서, 첨부된 도면에 개시된 특정 실시 예에 의해 기술적 사상이 제한되는 것은 아니며, 발명의 사상 및 기술 범위에 포함되는 모든 균등물 또는 대체물을 포함하는 것으로 이해되어야 한다.

[0024] 제1, 제2 등과 같이 서수를 포함하는 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 이러한 구성요소들은 상술한 용어에 의해 한정되지는 않는다. 상술한 용어는 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다.

[0025] 본 명세서에서, "포함한다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다. 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.

[0026] 한편, 본 명세서에서 사용되는 구성요소에 대한 "모듈" 또는 "부"는 적어도 하나의 기능 또는 동작을 수행한다. 그리고, "모듈" 또는 "부"는 하드웨어, 소프트웨어 또는 하드웨어와 소프트웨어의 조합에 의해 기능 또는 동작을 수행할 수 있다. 또한, 특정 하드웨어에서 수행되어야 하거나 적어도 하나의 프로세서에서 수행되는 "모듈" 또는 "부"를 제외한 복수의 "모듈들" 또는 복수의 "부들"은 적어도 하나의 모듈로 통합될 수도 있다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다.

- [0027] 그 밖에도, 본 발명을 설명함에 있어서, 관련된 공지 기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우, 그에 대한 상세한 설명은 축약하거나 생략한다.
- [0028] 도 2는 본 개시의 일 실시 예에 따른 연산 및 체크 포인팅을 수행하는 과정을 설명하는 도면이다.
- [0029] 도 2를 참조하면, 연산(computing)을 수행할 때 페이지 폴트(page fault)가 발생하면 파일 시스템 영역에 업데이트를 수행하는 전자 장치가 개시되어 있다. 본 개시는 비휘발성 메모리 중에서 뉴메모리를 사용한다. 뉴메모리(New Memory)는 비휘발성 특성을 가지고, 바이트 단위로 접근이 가능하며, 속도가 DRAM에 비해 크게 뒤쳐지지 않는 성능의 메모리를 의미한다. 예를 들어, 뉴메모리는 MRAM, FeRAM, RRAM, STT-MRAM, PRAM, 3D-XPoint 등을 포함할 수 있다. 뉴메모리는 종래의 주메모리와 스토리지의 역할을 수행할 수 있다. 따라서, 본 개시의 전자 장치는 주메모리와 스토리지를 별개로 사용하지 않아도 되는 장점이 있다. 또한, 주메모리와 스토리지를 하나의 뉴메모리로 사용하므로 전자 장치의 설계 공간적인 측면에서도 장점이 있다.
- [0030] 뉴메모리는 메인 메모리 영역과 파일 시스템 영역을 포함할 수 있다. 메인 메모리 영역은 연산 과정에서 사용되는 종래의 주메모리 역할을 수행할 수 있고, 파일 시스템 영역은 최종 데이터가 저장되는 종래의 스토리지 역할을 수행할 수 있다. 전자 장치는 연산을 수행할 때, 메인 메모리 영역의 데이터만을 업데이트할 수 있다. 그리고, 전자 장치는 특정 이벤트가 발생하는 경우, 또는 주기적으로 메인 메모리 영역의 데이터를 파일 시스템 영역으로 이동시킬 수 있다. 예를 들어, 특정 이벤트는 페이지 폴트가 발생하는 경우이다. 페이지 폴트는 메인 메모리 영역의 페이지가 오버플로우되거나 일정한 조건에 따라 페이지 내용을 초기화해야 하는 경우를 의미할 수 있다. 본 개시의 전자 장치는 일정한 조건에 따라 메인 메모리 영역의 데이터를 파일 시스템 영역으로 이동시키기 때문에 체크 포인트 시간이 줄어들 수 있다. 또한, 메인 메모리 영역과 파일 시스템 영역이 하나의 뉴메모리 내에 존재하므로 메인 메모리 영역으로부터 파일 시스템 영역으로 데이터를 이동시키는 시간이 적게 소요될 수 있다. 일 실시 예로서, 본 개시의 전자 장치의 연속 연산 시간(또는, checkpoint interval)은 10초 내외이고, 데이터를 이동시키는 페이지 폴트 및 체크 포인팅 시간은 전자 장치의 전체 처리 시간에 대비할 때 충분히 무시할 수 있는 시간이다.
- [0031] 그리고, 본 개시의 전자 장치는 수시로 메인 메모리 영역의 데이터를 파일 시스템 영역으로 이동시키기 때문에 오류가 발생하더라도 롤백(roll back) 시키기 위한 데이터가 상대적으로 적다. 따라서, 본 개시의 전자 장치는 롤백 시간이 종래의 전자 장치보다 현저히 줄어들 수 있다. 또한, 본 개시의 전자 장치는 다양한 방식을 적용하여 데이터 이동 시간을 줄이고, 데이터의 지속성(persistence) 및 일관성(consistency)을 보장할 수 있다. 구체적인 실시 예는 후술하기로 한다.
- [0032] 도 3은 본 개시의 일 실시 예에 따른 전자 장치의 블록도이다.
- [0033] 도 3을 참조하면, 전자 장치(100)는 프로세서(110)와 메모리(120)를 포함한다. 메모리(120)는 메인 메모리 영역과 파일 시스템 영역을 포함하는 뉴메모리다. 상술한 바와 같이, 메인 메모리 영역은 종래의 주메모리 역할을 수행하고, 파일 시스템 영역은 종래의 스토리지 역할을 수행할 수 있다.
- [0034] 메모리(120)의 메인 메모리 영역은 연산 과정에서 변경되는 데이터를 업데이트할 수 있다. 그리고, 메인 메모리 영역은 데이터가 변경된 페이지 정보를 라이팅하는 체크 포인트 테이블을 포함할 수 있다. 체크 포인트 테이블은 데이터가 변경된 페이지 정보 또는 복사된 페이지 정보를 포함할 수 있다. 본 개시의 일 실시 예로, 메모리(120)의 저장 단위를 페이지로 설명하고 있지만, 저장 단위는 바이트, 워드, 블록 등 다양하게 설정될 수 있다.
- [0035] 그리고, 메모리(120)는 파일 시스템 영역을 포함한다. 파일 시스템 영역은 기 설정된 이벤트 또는 기 설정된 주기에 따라 메인 메모리 영역의 데이터를 저장하는 역할을 수행한다. 파일 시스템 영역은 버퍼와 파일 저장부를 포함하고, 전자 장치(100)가 연산을 수행하는 동안 버퍼에 데이터를 저장해 두었다가, 연산이 종료되거나 업데이트 신호를 수신하면 버퍼의 데이터가 파일 저장부에 라이팅(writing)될 수 있다. 예를 들어, 연산이 종료되거나 업데이트 신호가 수신되는 경우는 파일 시스템 영역에 대한 체크 포인팅을 수행할 때일 수 있다.
- [0036] 프로세서(110)는 연산을 수행하고, 기 설정된 조건에 따라 메모리(120)의 메인 메모리 영역 및 파일 시스템 영역에 대해 체크 포인팅을 수행한다. 프로세서(110)는 메인 메모리 영역에 대해 체크 포인팅을 수행할 때, 체크 포인트 테이블을 초기화하고 메인 메모리 영역의 페이지의 속성을 읽기 전용(read-only)으로 설정할 수 있다.
- [0037] 또한, 프로세서(110)는 기 설정된 조건에 따라 연산 중이라도 메모리(120)의 메인 메모리 영역의 데이터를 정리하고 파일 시스템 영역으로 이동시킬 수 있다. 예를 들어, 프로세서(110)는 페이지에 데이터가 가득 찬 경우, 체크 포인트 테이블이 가득 찬 경우 등과 같이 페이지 폴트가 발생될 때 메인 메모리 영역의 데이터를 파일 시

스텝 영역으로 이동시킬 수 있다.

- [0038] 아래에서는 전자 장치(100)의 세부적인 블록도를 설명한다.
- [0039] 도 4는 본 개시의 일 실시 예에 따른 프로세서의 세부 블록도이다.
- [0040] 도 4를 참조하면, 프로세서(110)는 관리부(111), 제어부(112) 및 복원부(113)를 포함할 수 있다. 경우에 따라, 프로세서(110)는 관리부(111), 제어부(112) 또는 복원부(113) 중 일부 구성만을 포함할 수도 있다.
- [0041] 관리부(111)는 체크 포인팅이 수행되도록 지정된 이벤트가 발생하는지 여부를 판단할 수 있다. 일 실시 예로서, 관리부(111)는 프로세서(110)에 의해 수행되는 제1 프로세스에 관한 문맥 전환을 이벤트로서 판단할 수 있다. 아래의 설명에서 문맥 전환은 프로세서에 의해 수행되는 프로세스를 전환하기 위해 이전에 실행되는 프로세스의 상태를 저장하고, 새로운 프로세스 상태로 복구하는 전자 장치의 동작을 나타낼 수 있다. 제1 프로세스는 현재 진행 중인 연산 과정일 수 있다.
- [0042] 다른 일 실시 예로서, 관리부(111)는 프로세서(110)에 의해 수행되는 제1 프로세스에 관한 문맥 전환이 미리 지정된 횟수만큼 수행되는 경우를 이벤트로서 판단할 수 있다.
- [0043] 제어부(112)는 판단 결과에 따라, 이벤트가 발생한 시점의 제1 프로세스에 관한 정보를 뉴메모리 내에 저장할 수 있다. 보다 구체적으로, 제어부(112)는 제1 프로세스에 관한 어드레스를 롤 백(roll back)이 수행되기 위한 정보로서 저장할 수 있다. 또한, 제어부(112)는 fork() 함수 호출을 이용하여 제1 프로세스에 관한 태스크 구조를 뉴메모리로 복사할 수 있다. 제어부(112)는 제1 프로세스에 관한 제1 데이터의 변경이 발생하는 경우에, 태스크 구조 내에서 제1 데이터를 포함하는 페이지를 CoW(Copy on Write) 방식으로 업데이트할 수 있다.
- [0044] 또한, 제어부(112)는 제1 프로세스에 관한 최신의(latest) 어드레스를 뉴메모리 내에 롤백이 수행되기 위한 정보로서 저장하고, 이전에 저장된 나머지 어드레스를 삭제할 수 있다. 다른 일 실시 예로서, 제어부(112)는 제1 프로세스에 관한 최신의 어드레스를 버저닝(versioning)할 수 있다. 그에 따라, 제어부(112)는 복수의 시점에 각각 대응하는 복수의 버전의 복수의 어드레스를 뉴메모리 내에 저장할 수 있다.
- [0045] 다른 일 실시 예로서, 제어부(112)는 이벤트가 발생한 시점의 제1 데이터 및 제1 메타 데이터를 파일 시스템 내에 저장할 수 있다.
- [0046] 복원부(113)는 전자 장치에 관한 시스템 오류가 발생한 경우에, 메인 메모리 영역에 존재하는 제2 데이터 및 제2 메타 데이터를 삭제하고, 제1 데이터 및 제1 메타 데이터를 롤백할 수 있다. 예를 들어, 메인 메모리 영역은 프로세스 영역일 수 있다. 또한, 복원부(113)는 뉴메모리 내에 저장된 제1 프로세스에 관한 상태 정보를 이용하여 현재 실행 중인 제1 프로세스를 복원할 수 있다.
- [0047] 한편, 프로세서는 메모리의 관리를 위한 매니저를 포함할 수 있다. 그리고, 프로세서는 메인 메모리 영역을 관리하는 태스크 매니저 및 파일 시스템 영역을 관리하는 IO 매니저를 포함할 수 있다.
- [0048] 도 5는 본 개시의 일 실시 예에 따른 전자 장치의 아키텍처를 설명하는 도면이다.
- [0049] 도 5를 참조하면 매니저(10), P-task(11), P-io(12), 프로세스의 메모리 어드레스 공간(Memory Address Space of Process)(21) 및 프로세스의 파일(File of Process)(22)이 도시되어 있다. 매니저(10)는 P-task(11) 및 P-io(12)를 전체적으로 관리할 수 있다. P-task(11)는 태스크 매니저로서 메인 메모리 영역의 연산 및 체크 포인팅 등의 과정을 관리할 수 있다. 그리고, P-io(12)는 IO 매니저로서 파일 시스템 영역의 연산 및 체크 포인팅 등의 과정을 관리할 수 있다. 본 개시의 전자 장치는 메인 메모리 영역과 파일 시스템 영역을 포함하는 뉴메모리를 이용함으로써 데이터의 지속성을 유지할 수 있다. 따라서, 일 실시 예로서, 태스크 매니저는 P(persistent)-task(11)로 칭할 수 있고, IO 매니저는 P(persistent)-io로 칭할 수 있다.
- [0050] 체크 포인팅을 위해 메모리에 저장해야 하는 시스템 상태는 작업 영역과 관련된 정보, 즉, 프로세스의 메모리 주소 공간(21), 프로세스의 파일(22)을 포함할 수 있다. 메모리 주소 공간 및 파일 공간의 상태를 저장할 때 두 파일은 임의의 위치에 저장될 수 없지만, 전체 일관성 있는 상태로 동기화되어야 한다. 매니저(10)는 각 체크 포인팅 주기마다 P-task(11)와 P-io(12)간의 데이터를 동기화하여 모든 데이터가 일관되도록 한다. 예를 들어, 프로세스의 메모리 주소 공간(21)은 메인 메모리 영역일 수 있고, 파일 공간은 파일 시스템 영역일 수 있다.
- [0051] 구체적으로, P-task(11)(또는, 태스크 매니저)는 프로세스의 메모리 주소 공간(또는, 메인 메모리 영역)의 일관성을 유지 관리할 수 있다. 메모리 주소 공간에 대한 업데이트는 스토어(store) 명령어가 실행될 때만 발생된다. 따라서, 라이트(write) 연산이 업데이트로 호출하는 파일 영역(또는, 파일 시스템 영역)과는 다르다.

따라서, P-task(11)는 모든 스토어 명령에 대해 업데이트를 만들기 전에 원본 데이터의 복사본을 만든다. 일 실시 예로서, 메모리는 페이지 단위로 관리될 수 있으므로 페이지 단위로 사본이 만들어진다. 그러나, 모든 스토어 명령에서 사본이 생성되는 것은 아니다. 수정되지 않은 원본 데이터는 일시적인 오류로 인해 롤백할 데이터이다.

[0052] P-io(12)(또는, IO 매니저)는 파일 시스템(또는, 파일 시스템 영역)의 일관성을 유지 관리할 수 있다. 파일 시스템 영역은 버퍼와 파일 저장부를 포함할 수 있다. 즉, 파일에 대한 업데이트는 데이터 또는 메타데이터에 대해 라이트(write) 명령에 따라 발생된다. P-task(11)와 마찬가지로 P-io(12)는 연산과 체크 포인팅 단계를 번갈아 수행할 수 있다. 연산 단계의 모든 라이트 명령에 대해 원래 데이터의 복사본이 업데이트 전에 생성될 수 있다. 그러나, P-io(12)는 전통적인 파일 시스템에서 버퍼 캐시를 관리하는 것과 유사한 메커니즘을 사용할 수 있다. 라이팅이 발생되면, 수정된 블록이 버퍼에 저장되고 이 페이지 이후의 모든 리드(read) 및 라이트(write)는 버퍼에서 처리될 수 있다. 그러나, 버퍼의 내용은 제거되지 않고 체크 포인팅 수행 전까지 버퍼에 남는다. P-io(12)는 체크 포인트 단계에서 버퍼의 모든 페이지를 한꺼번에 파일 저장부로 플러시(flush)할 수 있다.

[0053] P-task(11)와 P-io(12)는 독립적인 구성으로 프로세스의 작업 영역(또는, 메인 메모리 영역)과 프로세스의 파일을 각각 검사할 수 있다. 매니저(10)는 P-task(11)와 P-io(12)를 동기화하여 데이터의 일관성을 유지하는 것이다. 그리고, 매니저(10)는 오류 발생시 복구 프로세스를 수행하는 것이다.

[0054] 매니저(10)가 P-task(11)와 P-io(12)를 동기화하는 방법에 대해 설명한다. P-task(11)와 P-io(12)는 연산과 체크 포인팅 과정을 번갈아 수행할 수 있다. 메인 메모리 영역의 연산 과정과 파일 시스템 영역의 연산 과정은 동시에 수행될 수 있다. 기 설정된 시점에 체크 포인팅이 시작된다. 체크 포인팅 간격은 사용자에게 의해 설정할 수 있다. 매니저(10)는 응용 프로그램과 관련된 프로세스를 중지시킨다. 응용 프로그램과 관련된 프로세스가 중지되면, 모든 프로세스의 전체 상태가 동기화될 수 있다. 다음에, P-task 체크 포인팅 단계가 수행되고 이후 P-io 체크 포인팅 단계가 수행될 수 있다. 기 설정된 체크 포인팅 순서는 오류 발생시 복구 프로세스에 의해 일관성 있는 상태로 응용 프로그램 상태를 반환하도록 할 수 있다.

[0055] 그리고, 매니저(10)는 복구 프로세스를 수행할 수 있다. 오류가 복구되면 전자 장치는 응용 프로그램을 일관된 상태로 되돌려야 한다. 매니저(10)는 저장된 응용 프로그램을 검사하고 복구 과정을 수행할 수 있다. 연산 단계 또는 P-task 체크 포인팅 단계에서 오류가 발생하는 경우, 응용 프로그램은 이전의 일관성 있는 상태로 복구될 수 있다. 즉, 데이터는 롤백 복구될 수 있다. 또는, P-io 체크 포인팅 단계에서 오류가 발생하는 경우, P-task 체크 포인팅 단계를 이미 통과하였으므로 데이터는 롤포워드될 수 있다. 즉, 본 개시의 전자 장치는 모든 파일 데이터가 버퍼에 있고, 파일 저장부에 버퍼에 있는 데이터를 쓰는 것은 멍등(idempotent)이기 때문에 파일 저장부에 버퍼 데이터를 쓰는 것만으로도 일관성을 유지할 수 있다.

[0056] 도 6은 본 개시의 일 실시 예에 따른 전자 장치가 프로세스에 관한 정보를 저장하는 과정을 설명하는 도면이다.

[0057] 도 6은 일 실시 예에 따른 전자 장치가 프로세스에 관한 정보를 저장하는 과정을 설명하는 예시도이다. 도 6을 참조하면, 미리 지정된 이벤트들(30, 40, 50)에 따라 프로세스에 관한 정보 P1 및 P2를 생성하여 저장하는 전자 장치가 제공될 수 있다.

[0058] 본 실시 예에서 프로세스에 관한 정보는 전체 시스템의 비정상적 종료와 같은 결함을 대비하여 롤백(roll back)이 수행되는 시점의 정보를 나타낼 수 있다. 일 실시 예에 따를 때, 미리 지정된 이벤트들(30, 40, 50)은 문맥 전환(context switching)으로 구현될 수 있다.

[0059] 도 6에서 도시된 것과 같이, 제1 문맥 전환(30)에 따라 제1 프로세스 A1에서 제2 프로세스 A2로 문맥이 전환될 수 있다. 마찬가지로, 제2 문맥 전환(30)에 따라 제3 프로세스 A3에서 다시 제1 프로세스 A1으로 문맥이 전환될 수 있다. 다만, 도 6에서 도시되는 이벤트들(30, 40, 50)의 실시 예는 본 발명의 사상의 이해를 돕기 위한 예시적 기재일 뿐, 다른 실시 예의 범위를 한정하거나 제한하는 것은 아니다. 이를 테면, 문맥 전환이 미리 지정된 횟수만큼 실행되는 시점이 이벤트로서 이용될 수도 있을 것이다. 구체적으로, 문맥 전환이 실행되는 커맨드가 n 배 호출된 시점이 이벤트로서 이용될 수 있다.

[0060] 전자 장치 내의 프로세서는 제1 문맥 전환(30)에 따라 제2 프로세스 A2를 새롭게 수행할 수 있다. 다만, 본 실시 예의 전자 장치는 시스템 에러와 같은 결함을 대비하여 롤백 시점으로 이용하기 위한 제1 프로세스 A1에 관한 정보 P1을 미리 저장할 수 있다. 보다 구체적으로, 전자 장치는 Fork() 함수를 이용하여 제1 프로세스 A1에 관한 정보 P1을 생성할 수 있다. 보다 구체적으로, 정보 P1 는 제1 문맥 전환(30)이 발생하는 시점의 제1 프로세스 A1에 관한 어드레스를 나타낼 수 있다.

- [0061] 또한, 전자 장치는 실행 중이었던 제1 프로세스 A1의 태스크 구조를 복제하는 방식으로 제1 프로세스 A1에 정보 P1을 생성할 수 있다. 그에 따라, 제1 프로세스 A1와 제1 프로세스 A1에 정보 P1은 동일한 메모리 공간을 공유하도록 생성될 수 있다.
- [0062] 전자 장치에 의해 다양한 프로세스가 실행되다가 제2 문맥 전환(40)에 의해 다시 제1 프로세스 A1가 실행되는 경우가 존재할 수 있다. 전자 장치는 제1 프로세스 A1에 관한 정보 P1으로부터 변경된 영역에 관한 페이지 정보만을 독립적으로 업데이트할 수 있다.
- [0063] 예시적으로, 제1 문맥 전환(30)이 수행된 시점의 제1 텍스트(61)가 2로 저장된 경우에, 제2 문맥 전환(40)에 따라 제1 텍스트(61)에 관한 변경이 발생된 경우가 존재할 수 있다. 전자 장치는 제1 텍스트(61)에 대해 Copy on Write 방식으로 제2 텍스트(62)로서 300이라는 데이터를 업데이트할 수 있다. 그에 따라, 추후에 제3 문맥 전환(50)이 발생하는 경우에, 전자 장치는 제1 프로세스 A1에 관한 정보 P2가 제2 텍스트(62)와 동일한 값에 대응하는 제3 텍스트(63)인 300을 포함하도록 생성할 수 있다.
- [0064] 본 실시 예에 따른 전자 장치는 Fork() 함수와 Copy on Write 방식을 이용하여 프로세스의 최근 상태를 체크 포인팅할 수 있다. 그에 따라, 본 실시 예에 따른 전자 장치는 시스템 전체의 부하를 줄이고, 가벼운 자원을 이용하는 것만으로 내구성을 높이는 효과를 기대할 수 있다.
- [0065] 도 7은 본 개시의 일 실시 예에 따른 전자 장치가 프로세스에 관한 정보를 관리하는 구조를 설명하는 도면이다.
- [0066] 도 7을 참조하면, 전자 장치에 포함되는 뉴메모리(120)는 실행중인 프로세스에 대해 런 큐(run queue)(121)와 퍼시스턴시 큐(persistency queue)(122)를 포함할 수 있다. 일 실시 예로서, 런 큐(121) 및 퍼시스턴시 큐(122)는 메모리(120)의 메인 메모리 영역에 포함될 수 있다.
- [0067] 전자 장치는 뉴메모리(120) 내에서 실행되는 프로세스들이 저장된 연결 리스트에 관한 런 큐(121)를 포함할 수 있다. 본 실시 예에서, 전자 장치는 런 큐(121)를 이용하여 제1 프로세스 A1, 제2 프로세스 A2 및 제3 프로세스 A3를 관리할 수 있다.
- [0068] 또한, 전자 장치는 현재 실행되지 않은 상태의 프로세스에 관한 정보를 저장하는 퍼시스턴시 큐(122)를 포함할 수 있다. 예시적으로, 제1 문맥 전환(30)에 대응하여 제1 프로세스 A1에 관한 정보 P1이 퍼시스턴시 큐(122)를 이용하여 관리될 수 있다. 제1 프로세스 A1에 관한 정보 P1은 제1 문맥 전환(30)이 발생한 시점의 제1 프로세스 A1의 상태에 관한 정보를 나타낼 수 있다. 예시적으로, 컴퓨팅 시스템 내에 시스템 오류와 같은 상황이 발생한 경우에, 제1 문맥 전환(30)이 발생한 시점은 프로세스가 롤백 되기 위한 기준 시점으로서 이용될 수 있다.
- [0069] 예시적으로, 런 큐(121)에 존재하는 제1 프로세스 A1이 실행되는 도중에 시스템의 결함이 발생한 경우가 있을 수 있다. 이 경우에, 전자 장치는 실행 중인 제1 프로세스 A1를 강제 종료 할 수 있다. 또한, 전자 장치는 퍼시스턴시 큐(122)에 미리 저장된 제1 프로세스 A1에 관한 정보 P1를 런 큐(121)로 복사하여 롤백 시점에 대응하는 제1 프로세스 A1를 다시 복원할 수 있다.
- [0070] 본 실시 예에 따른 전자 장치는 오늘날 뉴 메모리로서 널리 이용되는 NVRAMS의 비휘발성 특징을 이용하여 시스템 부하를 줄이면서 프로세스의 롤백 정보를 저장하여 내구성은 높이는 효과를 기대할 수 있다.
- [0071] 도 8은 본 개시의 다른 실시 예에 따른 전자 장치가 데이터 및 메타 데이터를 저장하는 과정을 설명하는 도면이다.
- [0072] 도 8을 참조하면, 본 실시 예에 따른 전자 장치에 포함되는 뉴메모리(120)의 구조도가 도시된다. 뉴메모리(120)는 메인 메모리 영역(121)과 파일 시스템 영역(122)을 포함할 수 있다.
- [0073] 본 실시 예에 따른 전자 장치는 실행 중인 프로세스에 따라 새로운 데이터의 생성(creating) 작업이나 수정(modification) 작업(531)이 실행되는 경우에 관련 데이터를 메인 메모리 영역(121)에 존재하는 제1 메타 데이터 M1에 업데이트 할 수 있다. 또한, 전자 장치는 실행 중인 프로세스에 따라 데이터의 쓰기(writing) 작업(73a, 73b)이 실행되는 경우에 관련 데이터를 메인 메모리 영역(121)에 존재하는 제1 데이터 D1에 업데이트 할 수 있다.
- [0074] 전자 장치는 체크 포인팅이 수행되는 미리 지정된 이벤트가 발생하는지 여부를 판단할 수 있다. 일 실시 예로서, 미리 지정된 이벤트는 전자 장치에 의해 수행되는 프로세스에 관한 문맥 전환을 나타낼 수 있다. 다른 일 실시 예로서, 미리 지정된 이벤트는 전자 장치에 의해 수행되는 프로세스에 관한 문맥 전환이 미리 지정된 횟수만큼 수행되는 경우를 나타낼 수 있다.

- [0075] 전자 장치는 미리 지정된 이벤트가 발생됨에 따라 메인 메모리 영역(121)에 존재하는 제1 데이터 D1 및 제1 메타 데이터 M1 모두를 파일 시스템 영역(122)으로 복사(73a, 73b)할 수 있다. 또한, 전자 장치는 복사된 데이터 및 메타 데이터를 발생된 이벤트의 시점에 대한 제2 데이터 D2 및 제2 메타 데이터 M2로서 저장할 수 있다. 본 실시 예에 따른 전자 장치는 미리 지정된 이벤트에 따라 데이터 및 메타 데이터 모두를 뉴메모리(120) 내의 파일 시스템 영역(122)으로 저장할 수 있다.
- [0076] 다만, 전자 장치는 제1 데이터 D1 및 제1 메타 데이터 M1가 파일 시스템 영역(122)으로 복사(73a, 73b)된 이후에도 데이터의 생성(creating) 작업이나 수정(modification) 작업(71) 또는 데이터의 쓰기(writing) 작업(72)이 발생하는 경우에 새로운 데이터 및 메타 데이터들을 메인 메모리 영역의 제1 데이터 D1 및 제1 메타 데이터 M1로서 지속적으로 업데이트할 수 있다.
- [0077] 프로세스를 실행하는 전자 장치에 시스템 오류나 결함이 발생한 경우를 가정하자. 전자 장치는 앞서 도 6에서 설명된 것과 같이 실행 중인 프로세스를 강제 종료하고, 프로세스에 관한 상태 정보를 롤백하여 복원할 수 있다. 본 실시 예에 따른 전자 장치는 메인 메모리 영역(121)에 저장된 제1 데이터 D1 및 제1 메타 데이터 M1를 삭제(74)할 수 있다. 또한, 전자 장치는 파일 시스템 영역(122) 내에 저장된 이전의 제2 데이터 D2 및 제2 메타 데이터 M2를 메인 메모리 영역(121)으로 다시 롤백시킬 수 있다. 그에 따라, 이전의 프로세스 시점에 대응하는 데이터 및 메타 데이터 역시도 함께 복원되어 시스템 전체의 내구성이 향상되는 효과를 기대할 수 있다.
- [0078] 도 9는 본 개시의 일 실시 예에 따른 연산 및 체크 포인팅을 수행하는 과정을 구체적으로 설명하는 도면이다.
- [0079] 도 9를 참조하면, 본 개시의 전자 장치가 연산 및 체크 포인팅을 수행하는 과정이 도시되어 있다. 전자 장치는 이전 단계의 연산이 종료되면 체크 포인팅을 하고, 다음 연산 과정을 수행한다.
- [0080] 전자 장치는 메인 메모리 영역에 대한 체크 포인팅을 수행할 수 있다. 전자 장치는 메인 메모리 영역에 대한 체크 포인팅을 수행할 때 체크 포인트 테이블을 초기화하고 메모리를 읽기 전용(read-only)으로 설정할 수 있다. 메모리가 읽기 전용으로 설정되었기 때문에 메인 메모리 영역의 페이지에는 데이터가 씌여질 수 없다. 체크 포인트 테이블은 메인 메모리 영역에 포함된 테이블로서, 데이터가 변경된 페이지 정보를 라이팅(writing)하는 역할을 수행한다. 본 실시 예에서는 데이터 단위를 페이지로 가정하고 설명하지만, 데이터 단위는 바이트, 워드, 2바이트, 2워드 등과 같이 다양하게 설정될 수 있다.
- [0081] 전자 장치는 페이지 10(80)과 관련한 연산을 수행할 수 있다. 연산 중에 페이지 10의 데이터가 업데이트될 수 있다. 그러나, 메모리의 속성이 읽기 전용이기 때문에 페이지 10(80)의 데이터는 변경되지 않는다. 전자 장치는 체크 포인트 테이블에 업데이트가 필요한 페이지 정보를 라이팅하고, 페이지 10을 복사한 페이지를 생성한다. 도 9에서는 페이지 14(80)가 이전의 페이지 10을 복사한 페이지이다. 그리고, 전자 장치는 체크 포인트 테이블에 변경된 페이지 정보를 라이팅한다. 도 9에 도시된 바와 같이 체크 포인트 테이블(86)은 페이지 10에서 페이지 14로 변경되었다는 정보를 포함한다.
- [0082] 전자 장치는 메모리의 속성을 쓰기(write)로 변경한다. 전자 장치는 페이지 10을 업데이트한다. 즉, 전자 장치는 업데이트된 페이지 10(80a)과 복사된 페이지 14(81)이 존재한다. 전자 장치는 상술한 과정을 통해 데이터를 업데이트할 수 있다. 또한, 전자 장치는 복사한 페이지 정보 및 데이터를 포함하고 있으므로 오류가 발생한 경우, 복사한 페이지 정보를 이용하여 용이하게 데이터를 롤백할 수 있다. 전자 장치는 메인 메모리 영역의 연산이 종료되면, 다시 메인 메모리 영역에 대해 체크 포인팅 과정을 수행한다.
- [0083] 한편, 상술한 바와 같이, 전자 장치는 메인 메모리 영역과 파일 시스템 영역의 데이터를 일치시켜야 한다. 파일 시스템 영역은 버퍼와 파일 저장부를 포함한다. 메인 메모리 영역의 데이터가 변경된 경우, 전자 장치는 파일 시스템 영역의 버퍼에 변경된 블록(92)을 라이팅한다. 연산 및 메인 메모리 영역에 대한 체크 포인팅 과정이 수행될 때, 전자 장치는 파일 저장부의 데이터 블록(91)은 유지한 채 버퍼에만 업데이트된 데이터 블록(92)을 가질 수 있다. 만일 연산 및 메인 메모리 영역에서 오류가 발생한 경우, 전자 장치는 버퍼에 저장된 업데이트된 데이터 블록(92)을 버리면 된다.
- [0084] 전자 장치는 메인 메모리 영역에 대한 체크 포인팅 과정이 종료되면, 파일 시스템 영역에 대한 체크 포인팅 과정을 수행한다. 파일 시스템 영역에 대한 체크 포인팅 과정을 수행할 때, 전자 장치는 버퍼에 저장된 데이터 블록(92)을 파일 저장부의 데이터 블록(91)에 라이팅한다. 만일, 파일 시스템 영역에 대해 체크 포인팅 과정 수행 중에 오류가 발생하더라도, 이미 메인 메모리 영역에 대한 체크 포인팅 과정은 종료되었으므로 전자 장치는 버퍼에 저장된 데이터 블록(92)을 파일 저장부에 라이팅할 수 있다. 즉, 전자 장치는 롤포워드할 수 있다.

- [0085] 즉, 전자 장치는 연산을 수행할 때 원본 페이지의 데이터 업데이트가 요청되는 경우 그 페이지의 정보를 체크 포인트 테이블에 라이팅할 수 있다. 그리고, 전자 장치는 그 페이지를 복사한 복사 페이지를 생성하고, 복사 페이지의 정보를 체크 포인트 테이블에 라이팅할 수 있다. 그리고, 전자 장치는 메모리의 속성을 쓰기로 변경하여 원본 페이지의 데이터를 업데이트할 수 있다. 또한, 전자 장치는 연산을 수행하거나 메인 메모리 영역에 대해 체크 포인팅을 수행할 때 오류가 발생하는 경우, 체크 포인트 테이블에 라이팅된 정보에 기초하여 원본 페이지를 롤백할 수 있다.
- [0086] 한편, 전자 장치는 파일 시스템 영역을 포함하고, 파일 시스템 영역은 버퍼 및 파일 저장부를 포함할 수 있다. 파일 저장부는 원본 데이터 블록을 포함할 수 있다. 전자 장치는 그 데이터 블록이 업데이트되는 경우, 업데이트된 데이터 블록을 버퍼에 저장할 수 있다. 그리고, 전자 장치는 파일 시스템 영역에 대해 체크 포인팅을 수행할 때 업데이트된 데이터 블록을 파일 저장부에 라이팅할 수 있다. 상술한 바와 같이, 전자 장치는 메인 메모리 영역에 대한 체크 포인팅을 수행한 후 파일 시스템 영역에 대한 체크 포인팅을 수행할 수 있다.
- [0087] 지금까지 데이터의 지속성 및 일관성을 유지시키기 위한 다양한 실시 예를 설명하였다. 아래에서는 전자 장치 제어 방법의 흐름도를 설명한다.
- [0088] 도 10은 본 개시의 일 실시 예에 따른 전자 장치 제어 방법의 흐름도이다.
- [0089] 도 10을 참조하면, 전자 장치는 체크 포인팅을 수행한다(S1010). 전자 장치의 뉴메모리는 메인 메모리 영역과 파일 시스템 영역을 포함한다. 전자 장치는 메인 메모리 영역에 대한 체크 포인팅을 수행한 후 파일 시스템 영역에 대한 체크 포인팅을 수행할 수 있다. 메인 메모리 영역은 데이터가 변경된 페이지 정보를 라이팅하는 체크 포인트 테이블을 포함할 수 있다. 그리고, 전자 장치는 메인 메모리 영역에 대해 체크 포인팅을 수행할 때 체크 포인트 테이블을 초기화하고, 메모리의 페이지 속성을 읽기 전용으로 설정할 수 있다.
- [0090] 전자 장치는 연산을 수행한다(S1020). 전자 장치는 원본 페이지의 데이터 업데이트가 요청되면, 원본 페이지 정보를 체크 포인트 테이블에 라이팅한다. 그리고, 전자 장치는 복사 페이지를 생성하고, 복사 페이지 정보도 체크 포인트 테이블에 라이팅한다. 전자 장치는 메모리 속성을 라이트로 변경하고 원본 페이지의 데이터를 업데이트할 수 있다.
- [0091] 도 11은 본 개시의 일 실시 예에 따른 이벤트에 따라 특정 프로세스에 관한 정보를 저장하는 전자 장치의 동작을 설명하는 흐름도이다.
- [0092] 도 11을 참조하면, 시스템의 내구성을 높이도록 동작하는 전자 장치의 체크 포인팅을 수행하는 방법이 도시된다. 전자 장치는 체크 포인팅이 수행되도록 지정된 이벤트가 발생하는지 여부를 판단할 수 있다(S1110). 체크 포인팅(check-pointing)은 실행 중인 프로세스에 관한 메인 메모리 영역 내에 저장된 변경된 데이터 및 메타 데이터들의 특정 시점에 대응하는 값을 저장하는 전자 장치의 동작을 나타낼 수 있다. 전자 장치는 미리 설정된 조건에 따라 다양한 이벤트들을 기준으로 하여 체크 포인팅을 수행할 수 있다.
- [0093] 전자 장치는, 판단 결과에 따라, 이벤트가 발생한 시점의 제1 프로세스에 관한 정보를 뉴메모리 내에 저장할 수 있다(S1120). 보다 구체적으로, 전자 장치는 이벤트가 발생한 시점의 제1 프로세스에 관한 어드레스를 뉴메모리 내에 저장할 수 있다. 또한, 전자 장치는 제1 프로세스에 관한 태스크(task) 구조를 뉴메모리 내에 복사할 수 있다.
- [0094] 도 12는 본 개시의 일 실시 예에 따른 이벤트에 따라 특정 데이터 및 메타 데이터를 저장하는 전자 장치의 동작을 설명하는 흐름도이다.
- [0095] 도 12를 참조하면, 체크 포인팅을 수행하는 전자 장치는 체크 포인팅이 수행되도록 지정된 이벤트가 발생하는지 여부를 판단할 수 있다(S1210). 판단의 결과에 따라, 전자 장치는 이벤트가 발생한 시점의 제1 데이터 및 제1 메타 데이터를 파일 시스템 내에 저장할 수 있다(S1220). 전자 장치는 이전 단계의 판단 결과에 따라 이벤트가 발생한 시점의 제1 데이터 및 제1 메타 데이터를 파일 시스템 내에 저장할 수 있다. 본 실시 예에서 메타 데이터는 디스크에 구조적으로 존재하는 데이터를 관리하기 위한 2차적 데이터를 나타낼 수 있다. 메타 데이터는 파일의 생성과 삭제, 디렉토리의 생성과 삭제, 파일 크기의 증가 감소에 따라 생성 또는 변경되는 데이터를 나타낼 수 있다. 본 실시 예에서 제1 데이터 및 제1 메타 데이터는 이벤트가 발생한 시점에 실행되는 프로세스에 연관되는 데이터 및 메타 데이터 각각을 나타낼 수 있다.
- [0096] 전자 장치에 관한 시스템 오류가 발생한 경우에, 메인 메모리 영역에 존재하는 제2 데이터 및 제2 메타 데이터를 삭제하고, 제1 데이터 및 제1 메타 데이터를 롤백할 수 있다(S1230). 전자 장치는 시스템 오류 또는 결함의

경우에 뉴메모리의 파일 시스템 내에 저장된 제1 데이터 및 제1 메타 데이터를 이용하여 롤백을 수행할 수 있다. 보다 구체적으로, 전자 장치는 시스템 오류가 발생한 경우에 메인 메모리 영역에 존재하는 제2 데이터 및 제2 메타 데이터를 삭제할 수 있다. 이에 따라, 전자 장치는 파일 시스템 내에서 저장되어 있는 이전의 제1 데이터 및 제1 메타 데이터를 롤백하여 이용할 수 있다.

[0097] 이상에서 설명된 실시 예들은 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시 예들에서 설명된 장치, 방법 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0098] 소프트웨어는 컴퓨터 프로그램(computer <0055> program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 비일시적 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0099] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

[0100] 또한, 이상에서는 본 발명의 바람직한 실시 예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특정의 실시 예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전망으로부터 개별적으로 이해되어져서는 안될 것이다.

부호의 설명

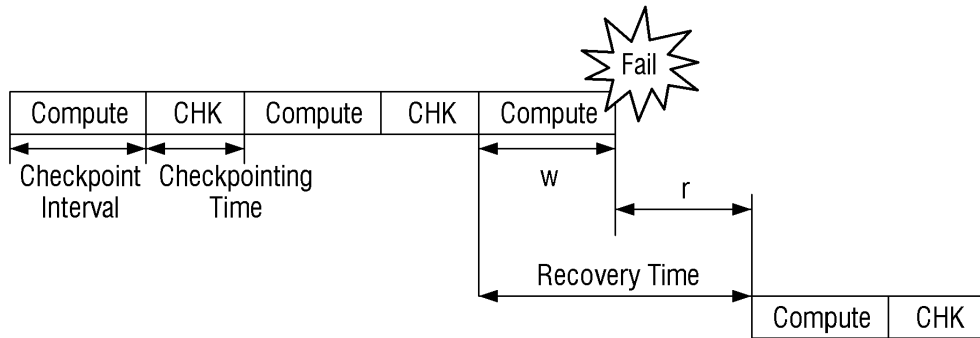
[0101] 100: 전자 장치

110: 프로세서

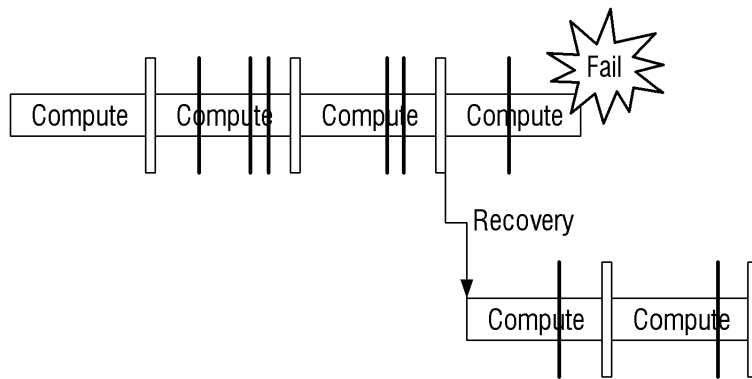
120: 메모리

도면

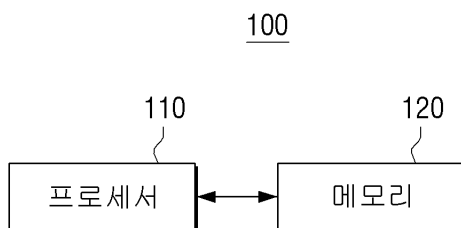
도면1



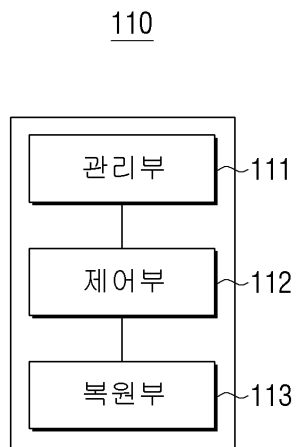
도면2



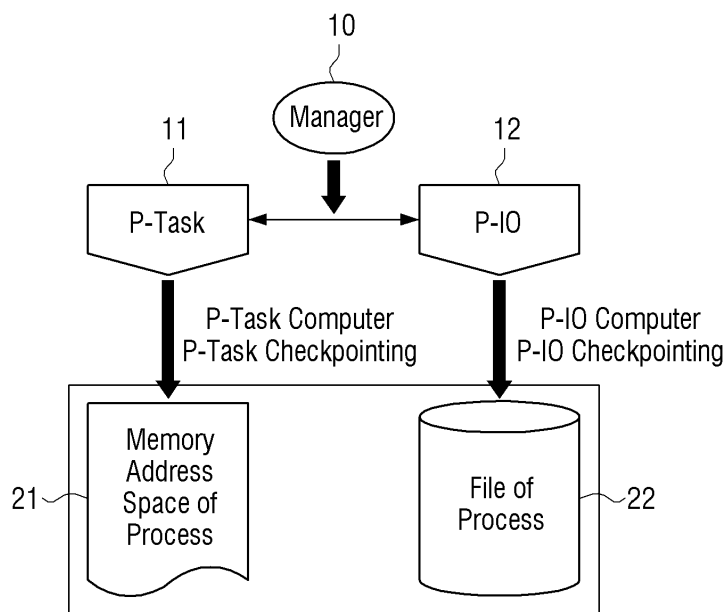
도면3



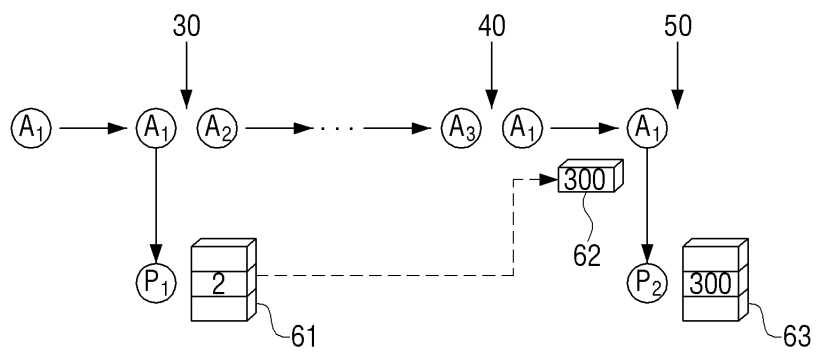
도면4



도면5

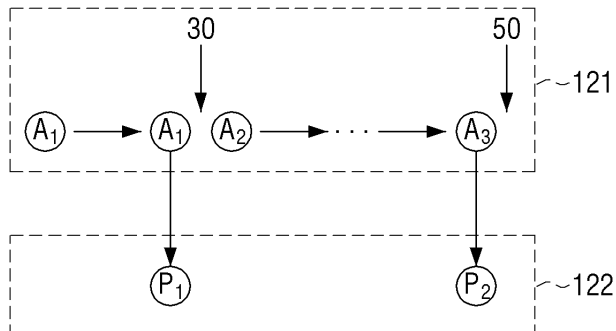


도면6



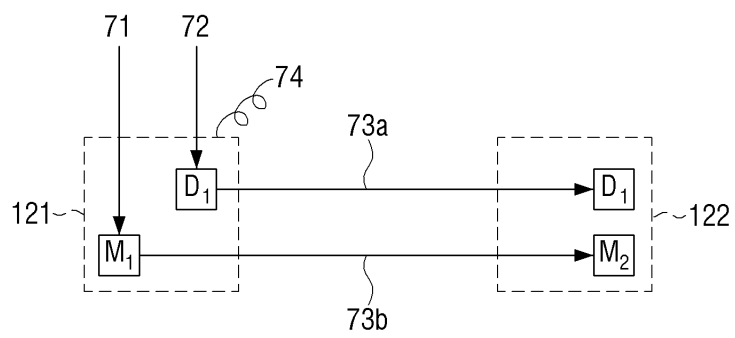
도면7

120

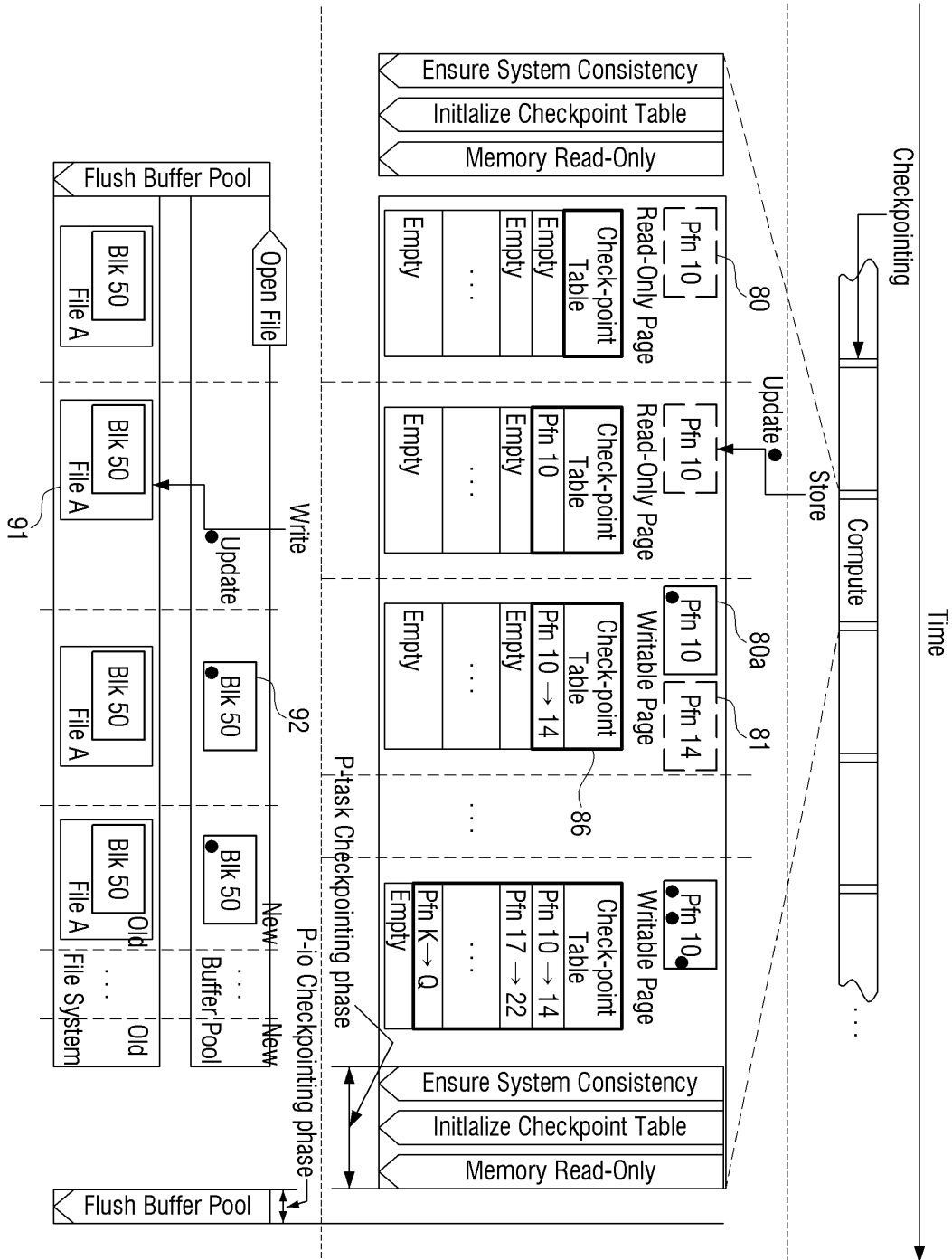


도면8

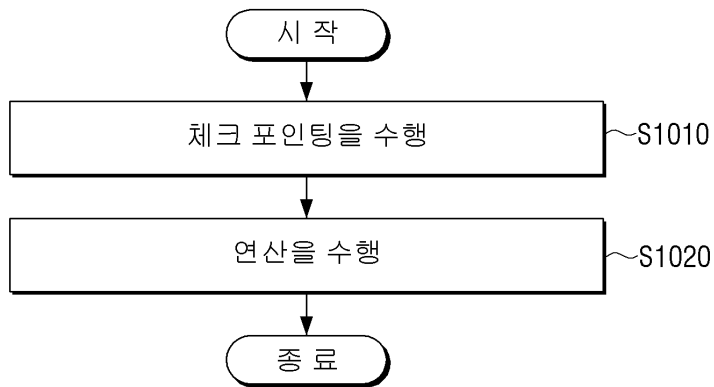
120



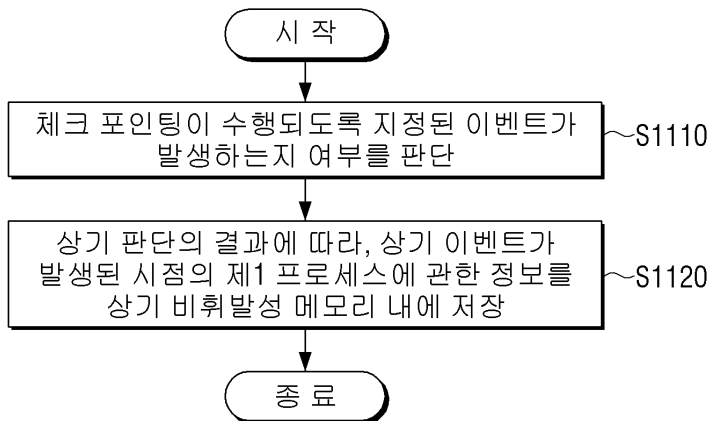
도면9



도면10



도면11



도면12

