



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0101307
(43) 공개일자 2017년09월05일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 15/17 (2006.01) G06F 15/173 (2006.01)
G06F 9/38 (2006.01) G06F 9/54 (2006.01)</p> <p>(52) CPC특허분류
G06F 15/17 (2013.01)
G06F 15/173 (2013.01)</p> <p>(21) 출원번호 10-2017-7023002</p> <p>(22) 출원일자(국제) 2015년12월28일
심사청구일자 2017년08월18일</p> <p>(85) 번역문제출일자 2017년08월18일</p> <p>(86) 국제출원번호 PCT/JP2015/006491</p> <p>(87) 국제공개번호 WO 2016/132416
국제공개일자 2016년08월25일</p> <p>(30) 우선권주장
JP-P-2015-030566 2015년02월19일 일본(JP)</p> | <p>(71) 출원인
닛본 덴끼 가부시끼가이샤
일본국 도쿄도 미나토구 시바 5쥬메 7방 1코</p> <p>(72) 발명자
가와하라, 아오이
일본 1088001 도쿄도 미나토구 시바 5쥬메 7-1 닛본 덴끼 가부시끼가이샤 내</p> <p>(74) 대리인
양영준, 박충범</p> |
|---|---|

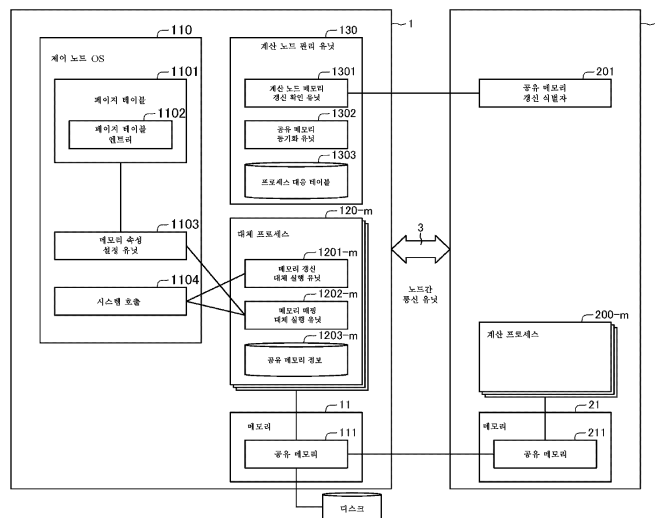
전체 청구항 수 : 총 13 항

(54) 발명의 명칭 정보 처리 장치, 정보 처리 방법, 메인 프로세서 코어, 프로그램, 정보 처리 방법 및 서브 프로세서 코어

(57) 요약

제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어가 포함된다. 운영 체제는 메인 프로세서 코어에 통합되어 있고 운영 체제는 서브 프로세서 코어에 통합되어 있지 않다. 공유 메모리들은 제1 및 제2 메모리들에 각각 형성된다. 제1 및 제2 메모리들의 공유 메모리 내의 데이터는 동기화된다. 메인 프로세서 코어는, 서브 프로세서 코어가 동작을 정지하는 동안, 제1 및 제2 메모리들에 형성된 공유 메모리 영역들 내의 데이터를 동기화하도록 구성된다.

대표도



(52) CPC특허분류

G06F 9/38 (2013.01)

G06F 9/54 (2013.01)

명세서

청구범위

청구항 1

제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 상기 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어를 포함하는 정보 처리 장치로서- 상기 메인 프로세서 코어에는 운영 체제가 통합되어 있고, 상기 서브 프로세서 코어에는 운영 체제가 통합되어 있지 않음 -,

상기 제1 메모리에 공유 메모리 영역이 형성되고, 상기 제2 메모리에 공유 메모리 영역이 형성되고, 상기 제1 및 제2 메모리들의 상기 공유 메모리 영역들 내의 데이터가 동기화되고;

상기 메인 프로세서 코어는, 상기 서브 프로세서 코어가 동작을 정지하는 동안, 상기 제1 및 제2 메모리들에 형성된 상기 공유 메모리 영역들 내의 상기 데이터를 동기화하도록 구성된, 정보 처리 장치.

청구항 2

제1항에 있어서,

상기 서브 프로세서 코어는, 상기 메인 프로세서 코어에 미리 결정된 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하도록 구성되고;

상기 메인 프로세서 코어는 상기 서브 프로세서 코어에 의해 요청된 상기 미리 결정된 정보 처리를 실행한 후, 상기 공유 메모리 영역들 중 하나에서 상기 미리 결정된 정보 처리를 통해 갱신된 데이터를 상기 공유 메모리 영역들 중 다른 하나로 복사하고, 그 후 상기 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하도록 구성된, 정보 처리 장치.

청구항 3

제1항 또는 제2항에 있어서,

상기 서브 프로세서 코어는, 상기 메인 프로세서 코어에 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하도록 구성되고- 상기 정보 처리는 상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 데이터를 갱신하는 것임 -, 및

상기 메인 프로세서 코어는 상기 서브 프로세서 코어에 의해 요청된 상기 정보 처리를 실행한 후, 상기 정보 처리를 통해 갱신된, 상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 상기 데이터를 상기 제1 메모리에 형성된 상기 공유 메모리 영역으로 복사하고, 그 후 상기 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하도록 구성된- 상기 정보 처리는 상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 상기 데이터를 갱신하는 것임 -, 정보 처리 장치.

청구항 4

제2항 또는 제3항에 있어서,

상기 정보 처리는 시스템 호출인, 정보 처리 장치.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 메인 프로세서 코어는, 상기 서브 프로세서 코어가 동작을 정지할 때, 상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 데이터가 갱신되어 있는지 여부를 확인하고, 상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 상기 갱신 데이터를 상기 제1 메모리에 형성된 상기 공유 메모리 영역으로 복사하도록 구성된, 정보 처리 장치.

청구항 6

제1항 내지 제5항 중 어느 한 항에 있어서,

상기 메인 프로세서 코어는, 상기 서브 프로세서 코어가 동작을 정지할 때, 상기 제1 메모리에 형성된 상기 공유 메모리 영역 내의 갱신 데이터를 상기 제2 메모리에 형성된 상기 공유 메모리 영역으로 복사하도록 구성된, 정보 처리 장치.

청구항 7

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 서브 프로세서 코어는, 상기 메인 프로세서 코어에 공유 메모리 영역을 형성하도록 요청하고, 또한 동작을 정지하도록 구성되고;

상기 메인 프로세서 코어는, 상기 서브 프로세서 코어에 의한 요청에 응답하여, 상기 제1 및 제2 메모리들 내에 상기 공유 메모리 영역들을 형성하고, 그 후 상기 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하도록 구성된, 정보 처리 장치.

청구항 8

제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 상기 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어를 포함하는 정보 처리 장치에 의해 실행되는 정보 처리 방법으로서- 상기 메인 프로세서 코어에는 운영 체제가 통합되어 있고, 상기 서브 프로세서 코어에는 운영 체제가 통합되어 있지 않고, 상기 제1 및 제2 메모리들 각각에 공유 메모리 영역들이 형성되고, 상기 제1 및 제2 메모리들의 상기 공유 메모리 영역들 내의 데이터는 동기화됨 -, 상기 정보 처리 방법은,

상기 메인 프로세서 코어에 의해, 상기 서브 프로세서 코어가 동작을 정지하는 동안, 상기 제1 및 제2 메모리들에 형성된 상기 공유 메모리 영역들 내의 상기 데이터를 동기화하는 단계를 포함하는, 정보 처리 방법.

청구항 9

제8항에 있어서,

상기 서브 프로세서 코어에 의해, 상기 메인 프로세서 코어에 미리 결정된 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하는 단계; 및

상기 메인 프로세서 코어에 의해, 상기 서브 프로세서 코어에 의해 요청된 상기 미리 결정된 정보 처리를 실행한 후, 상기 공유 메모리 영역들 중 하나에서 상기 미리 결정된 정보 처리를 통해 갱신된 데이터를 상기 공유 메모리 영역들 중 다른 하나로 복사하고, 그 후 상기 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하는 단계를 포함하는, 정보 처리 방법.

청구항 10

운영 체제가 통합되어 있는 메인 프로세서 코어로서- 상기 메인 프로세서 코어는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어를 제어하기 위한 기능을 가짐 -,

제1 메모리가 포함되고;

상기 제1 메모리에 공유 메모리 영역이 형성되고, 상기 공유 메모리 영역 내의 데이터는 상기 서브 프로세서 코어에 의해 포함된 제2 메모리 내의 데이터와 동기화되고;

공유 메모리 동기화 수단이 포함되고, 상기 공유 메모리 동기화 수단은 상기 서브 프로세서 코어가 동작을 정지하는 동안 상기 제1 메모리에 형성된 상기 공유 메모리 영역 내의 상기 데이터를 상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 데이터와 동기화하도록 구성된, 메인 프로세서 코어.

청구항 11

메인 프로세서 코어로 하여금 공유 메모리 동기화 수단으로서 기능하게 하기 위한 명령어들을 포함하는 프로그램으로서,

상기 메인 프로세서 코어는 제1 메모리를 갖고, 서브 프로세서 코어를 제어하기 위한 기능을 갖고, 공유 메모리 영역이 상기 제1 메모리에 형성되고, 상기 공유 메모리 내의 데이터는 상기 서브 프로세서 코어에 의해 포함되

는 제2 메모리의 데이터와 동기화되고, 운영 체제가 상기 메인 프로세서 코어에 통합되어 있고, 운영 체제가 상기 서브 프로세서 코어에 통합되어 있지 않고;

상기 공유 메모리 동기화 수단은 상기 서브 프로세서 코어가 동작을 정지하는 동안 상기 제1 메모리에 형성된 상기 공유 메모리 영역 내의 상기 데이터를 상기 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하도록 구성된, 프로그램.

청구항 12

운영 체제가 통합되어 있는 메인 프로세서 코어에 의해 실행되는 정보 처리 방법으로서- 상기 메인 프로세서 코어는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어를 제어하기 위한 기능을 가진 -, 상기 정보 처리 방법은,

상기 서브 프로세서 코어가 동작을 정지하는 동안 상기 메인 프로세서 코어에 의해 포함된 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 상기 서브 프로세서 코어에 의해 포함된 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하는 단계를 포함하는 정보 처리 방법.

청구항 13

운영 체제가 통합되어 있는 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어로서- 상기 서브 프로세서 코어에 운영 체제가 통합되어 있지 않음 -,

제2 메모리가 포함되고;

상기 제2 메모리에 공유 메모리 영역이 형성되고, 상기 공유 메모리 영역 내의 데이터는 상기 메인 프로세서 코어에 의해 포함된 제1 메모리 내의 데이터와 동기화되고;

상기 제2 메모리에 형성된 상기 공유 메모리 영역 내의 상기 데이터는 상기 서브 프로세서 코어가 동작을 정지하는 동안 상기 제1 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화되는, 서브 프로세서 코어.

발명의 설명

기술 분야

[0001] 본 발명은, 정보 처리 장치, 정보 처리 방법, 메인 프로세서 코어, 프로그램, 정보 처리 방법 및 서브 프로세서 코어에 관한 것이다.

배경 기술

[0002] 제어 노드와 연산 노드(계산 노드)를 포함하는 컴퓨터가 알려져 있다(예를 들어, 특허문헌 1). 특허문헌 1에 기재된 기술에서는, 연산 노드가 제어 노드에 의해 할당된 리모트 처리를 실행한다. 또한, 연산 노드에 의해 실행되는 리모트 처리에 의해 발생하는 입력/출력은, 제어 노드를 개재해서 행해지게 된다.

[0003] 또한, 전술한 바와 같이 복수의 노드를 포함할 경우, 노드들 사이(프로그램들 사이)에서 데이터를 교환하기 위해서 공유 메모리들이 사용된다. 이 경우, 공유 메모리들 사이의 데이터의 동기화의 제어가 요구된다. 이러한 공유 메모리들 사이의 데이터 동기화의 제어를 위한 기술로서, 예를 들어 특허문헌들 2 및 3이 알려져 있다. 특허문헌 2는, 중앙 처리 유닛의 각자의 메모리 버스에 설치된 공유 메모리들을 동기화하기 위한 방법을 설명한다. 특허문헌 3은, 전송 장치에 설치된 공유 메모리의 데이터를 주기적으로 송신함으로써 데이터를 매칭하기 위해 사용되는 방법을 설명한다.

선행기술문헌

특허문헌

- [0004] (특허문헌 0001) 일본 미심사 특허 출원 공보 제JP-A 2008-165318호
- (특허문헌 0002) 일본 미심사 특허 출원 공보 제JP-A H3-268158호(1991)

(특허문헌 0003) 일본 미심사 특허 출원 공보 제JP-A H9-269934호(1997)

발명의 내용

- [0005] 전술한 바와 같은 제어 노드와 계산 노드를 포함하는 컴퓨터에서, 계산 노드의 성능을 나타내기 위해서, 계산 노드가 정확한 예외(precise exception) 기능을 갖지 않을 경우가 있다. 이러한 경우, 계산 노드는, 예외가 발생한 순간의 명령어에서 정지하고, 예외 핸들링(handling) 후에 그 정지한 명령어로부터 처리를 복귀할 수 없다. 그 결과, 공간 부재 예외로 동작하는 메모리 매핑된 파일(memory mapped file) 기능을 사용하는 것이 곤란해지고, 상이한 프로세스들 사이에서 메모리 공유를 행하는 것이 곤란해질 수 있다.
- [0006] 즉, 일반적인 메모리 매핑된 파일에서는, 공간 부재 예외 핸들링에서 페이징이 실행되고, 실제로 메모리에 파일의 콘텐츠가 생성된다. 따라서, 메모리 매핑된 파일을 정확한 예외가 가능한 CPU(Central Processing Unit)에서 실행하는 것이 전제가 되고 있었다. 그 결과, 정확한 예외를 지원하지 않는 CPU를 갖는 계산 노드에서는, 메모리 매핑된 파일 기능을 사용하는 것이 곤란하고, 상이한 프로세스들 사이에서 메모리 공유를 행하는 것이 곤란하다는 문제가 있었다.
- [0007] 전술한 바와 같이, 정확한 예외를 지원하지 않는 CPU를 갖는 계산 노드에서, 메모리 매핑된 파일 기능을 사용하는 것이 불가능하고, 상이한 프로세스들 사이에서 메모리 공유를 행하는 것이 곤란하다는 문제가 있었다.
- [0008] SUMMARY
- [0009] 따라서, 본 발명의 목적은, 정확한 예외를 지원하지 않는 CPU를 갖는 계산 노드에서, 메모리 매핑된 파일 기능을 사용하는 것이 불가능하고, 상이한 프로세스들 사이에서 메모리 공유를 행하는 것이 곤란하다는 문제를 해결하는 정보 처리 장치를 제공하는 것이다.
- [0010] 이러한 목적을 달성하기 위해, 본 발명의 양태로서 정보 처리 장치는, 제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어를 포함한다. 운영 체제는 메인 프로세서 코어에 통합되어 있다. 운영 체제는 서브 프로세서 코어에 통합되어 있지 않다.
- [0011] 공유 메모리 영역은 제1 메모리에 형성되고, 공유 메모리는 제2 메모리에 형성되고, 제1 및 제2 메모리들의 공유 메모리 영역들 내의 데이터가 동기화된다.
- [0012] 메인 프로세서 코어는, 서브 프로세서 코어가 동작을 정지하는 동안, 제1 및 제2 메모리들에 형성된 공유 메모리 영역들 내의 데이터를 동기화하도록 구성된다.
- [0013] 또한, 본 발명의 다른 양태로서 정보 처리 방법은, 제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어를 포함하는 정보 처리 장치에 의해 실행되는 정보 처리 방법이다. 운영 체제는 메인 프로세서 코어에 통합되어 있다. 운영 체제는 서브 프로세서 코어에 통합되어 있지 않다. 공유 메모리 영역들은 각각 제1 및 제2 메모리들에 형성된다. 제1 및 제2 메모리들의 공유 메모리 영역들 내의 데이터가 동기화된다.
- [0014] 정보 처리 방법은, 메인 프로세서 코어에 의해, 서브 프로세서 코어가 동작을 정지하는 동안, 제1 및 제2 메모리들에 형성된 공유 메모리 영역들 내의 데이터를 동기화하는 단계를 포함한다.
- [0015] 또한, 본 발명의 다른 양태로서 메인 프로세서 코어는, 운영 체제가 통합되어 있는 메인 프로세서 코어이다. 메인 프로세서 코어는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어를 제어하기 위한 기능을 갖는다.
- [0016] 제1 메모리가 포함된다.
- [0017] 공유 메모리 영역은 제1 메모리에 형성되고, 공유 메모리 영역 내의 데이터는 서브 프로세서 코어에 포함된 제2 메모리의 데이터와 동기화된다.
- [0018] 공유 메모리 동기화 수단이 포함되고, 공유 메모리 동기화 수단은 서브 프로세서 코어가 동작을 정지하는 동안 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하도록 구성된다.
- [0019] 또한, 본 발명의 다른 양태로서 프로그램은, 메인 프로세서 코어로 하여금 공유 메모리 동기화 수단으로서 기능하게 하기 위한 명령어들을 포함하는 프로그램이다.

- [0020] 메인 프로세서 코어는 제1 메모리를 갖고, 서브 프로세서 코어를 제어하기 위한 기능을 갖고, 공유 메모리 영역이 제1 메모리에 형성되고, 공유 메모리 내의 데이터는 서브 프로세서 코어에 의해 포함되는 제2 메모리의 데이터와 동기화되고, 운영 체제가 메인 프로세서 코어에 통합되어 있고 운영 체제가 서브 프로세서 코어에 통합되어 있지 않다.
- [0021] 공유 메모리 동기화 수단은 서브 프로세서 코어가 동작을 정지하는 동안 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하도록 구성된다.
- [0022] 또한, 본 발명의 다른 양태로서 정보 처리 방법은, 운영 체제가 통합되어 있는 메인 프로세서 코어에 의해 실행되는 정보 처리 방법이다. 메인 프로세서 코어는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어를 제어하기 위한 기능을 갖는다.
- [0023] 정보 처리 방법은, 서브 프로세서 코어가 동작을 정지하는 동안 메인 프로세서 코어에 의해 포함된 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 서브 프로세서 코어에 의해 포함된 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하는 단계를 포함한다.
- [0024] 또한, 본 발명의 다른 양태로서 서브 프로세서 코어는, 운영 체제가 통합되어 있는 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어이다. 운영 체제는 서브 프로세서 코어에 통합되어 있지 않다.
- [0025] 제2 메모리가 포함된다.
- [0026] 공유 메모리 영역은 제2 메모리에 형성되고, 공유 메모리 영역 내의 데이터는 메인 프로세서 코어에 의해 포함된 제1 메모리의 데이터와 동기화된다.
- [0027] 제2 메모리에 형성된 공유 메모리 영역 내의 데이터는 서브 프로세서 코어가 동작을 정지하는 동안 제1 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화된다.
- [0028] 본 발명은, 전술한 바와 같은 구성들에 의해, 정확한 예외를 지원하지 않는 CPU를 갖는 계산 노드에서는, 메모리 매핑된 파일 기능을 사용하는 것이 불가능하고, 상이한 프로세스들 사이에서 메모리 공유를 행하는 것이 곤란하다는 문제를 해결하는 정보 처리 장치를 제공할 수 있다.

도면의 간단한 설명

- [0029] 도 1은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치의 구성의 일례를 도시하는 블록도이다;
- 도 2는 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 코어의 구성의 일례를 도시하는 블록도이다;
- 도 3은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에 의한 처리를 설명하기 위한 도면이다;
- 도 4는 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서, 제어 노드 및 계산 노드의 구성의 일례를 도시하는 블록도이다;
- 도 5는 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 노드가 계산 프로세스를 실행할 때의 구성의 일례를 도시하는 블록도이다;
- 도 6은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 사용되는 프로세스 대응 테이블의 일례를 도시하는 도면이다;
- 도 7은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 프로세스가 공유 메모리를 취득할 때의 계산 노드의 동작의 일례를 도시하는 흐름도이다;
- 도 8은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 프로세스가 공유 메모리를 취득할 때 제어 노드의 동작의 일례를 도시하는 흐름도이다;
- 도 9는 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 프로세스가 시스템 호출을 실행했을 때 공유 메모리들의 동기화(계산 노드 측)의 일례를 도시하는 흐름도이다;
- 도 10은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 프로세스가 시스템 호출을 실행했을 때 공유 메모리들의 동기화(제어 노드 측)의 일례를 도시하는 흐름도이다;
- 도 11은 계산 프로세스가 시스템 호출을 실행했을 때 공유 메모리들의 동기화의 흐름의 일례를 도시하는 도면이

다;

도 12는 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 계산 프로세스가 CPU 명령어에 의해 메모리 갱신을 행했을 때 동기화의 일례를 도시하는 흐름도이다;

도 13은 계산 프로세스가 CPU 명령어에 의해 메모리 갱신을 행했을 때 동기화의 흐름의 일례를 도시하는 도면이다;

도 14는 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 제어 노드의 프로세스가 시스템 호출을 실행하고 공유 메모리를 갱신했을 때 동기화의 일례를 도시하는 흐름도이다;

도 15는 제어 노드의 프로세스가 시스템 호출을 실행하고 공유 메모리를 갱신했을 때 동기화의 흐름의 일례를 도시하는 도면이다;

도 16은 본 발명의 제1 예시적인 실시예에 따른 정보 처리 장치에서 제어 노드의 프로세스가 CPU 명령어에 의해 공유 메모리를 갱신했을 때 동기화의 일례를 도시하는 흐름도이다;

도 17은 제어 노드의 프로세스가 CPU 명령어에 의해 공유 메모리를 갱신했을 때 동기화의 일례를 도시하는 도면이다;

도 18은 본 발명의 제2 예시적인 실시예에 따른 정보 처리 장치의 구성을 개략적으로 도시하는 개략적인 블록도이다; 그리고

도 19는 본 발명의 제2 예시적인 실시예에 따른 정보 처리 장치의 다른 구성을 개략적으로 도시하는 개략적인 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0030] 예시적인 실시예들
- [0031] [제1 예시적인 실시예]
- [0032] 본 발명의 제1 예시적인 실시예가 도 1 내지 도 17을 참조하여 설명될 것이다. 도 1은 정보 처리 장치의 구성의 일례를 도시하는 블록도이다. 도 2는 계산 코어(24)의 구성의 일례를 도시하는 블록도이다. 도 3은 정보 처리 장치에 의해 처리를 설명하기 위한 도면이다. 도 4는 제어 노드(1) 및 계산 노드(2)의 구성의 일례를 도시하는 블록도이다. 도 5는 계산 노드(2)가 계산 프로세스 wp를 실행할 때 구성의 일례를 도시하는 도면이다. 도 6은 프로세스 대응 테이블(1303)의 일례를 도시하는 도면이다. 도 7 및 도 8은 계산 프로세스가 공유 메모리를 취득할 때 동작 일례를 도시하는 흐름도이다. 도 9 및 도 10은 계산 프로세스가 시스템 호출을 실행했을 때 공유 메모리들의 동기화의 일례를 도시하는 흐름도이다. 도 11은 계산 프로세스가 시스템 호출을 실행했을 때 공유 메모리들의 동기화의 흐름의 일례를 도시하는 도면이다. 도 12는 계산 프로세스가 CPU 명령어에 의해 메모리 갱신을 행했을 때 동기화의 일례를 도시하는 흐름도이다. 도 13은 계산 프로세스가 CPU 명령어에 의해 메모리 갱신을 행했을 때 동기화의 흐름의 일례를 도시하는 도면이다. 도 14는 제어 노드에서의 프로세스가 시스템 호출을 실행하고 공유 메모리를 갱신했을 때 동기화의 일례를 도시하는 흐름도이다. 도 15는 제어 노드에서의 프로세스가 시스템 호출을 실행하고 공유 메모리를 갱신했을 때 동기화의 흐름의 일례를 도시하는 도면이다. 도 16은 제어 노드에서의 프로세스가 CPU 명령어에 의해 공유 메모리를 갱신했을 때 동기화의 일례를 도시하는 흐름도이다. 도 17은 제어 노드에서의 프로세스가 CPU 명령어에 의해 공유 메모리를 갱신했을 때 동기화의 일례를 도시하는 도면이다.
- [0033] 도 1을 참조하면, 본 예시적인 실시예에서 정보 처리 장치는 메인 프로세서로서 노드(1) 및, 서브 프로세서로서 노드(2)를 갖는다. 또한, 노드(1) 및 노드(2)는 노드간 통신 수단(3)을 개재해서 서로 통신할 수 있도록 구성된다. 본 예시적인 실시예에서 정보 처리 장치는 예를 들어, HPC(High Performance Computing)를 위한 병렬 컴퓨터로서 이용된다.
- [0034] 또한, 본 예시적인 실시예에서는, 정보 처리 장치가 2개의 노드, 노드(1) 및 노드(2)를 포함하는 경우가 설명될 것이다. 그러나, 정보 처리 장치의 구성은 이러한 경우에 한정되지 않는다. 즉, 정보 처리 장치는, 3개 이상의 노드를 포함할 수 있다. 또한, 본 예시적인 실시예에서는, 노드(1) 및 노드(2)는 동일한 컴퓨터 아키텍처를 갖는다. 그러나, 노드(1) 및 노드(2)는 상이한 컴퓨터 아키텍처를 사용할 수 있다.
- [0035] 도 1에서 도시된 바와 같이, 노드(1, 2)는 메모리(11, 21), CPU(12, 22) 및 데이터 전송 수단들(13, 23)을 갖

는다. CPU(12, 22)는 m개(m은 1 이상의 양의 정수를 나타냄)의 프로세서 코어(14, 24)를 갖는다. 메모리(11, 21), CPU(12, 22) 및 데이터 전송 수단들(13, 23)은 노드 내에서는 PCI 익스프레스 등에 의해 접속되어 있고, 프로세서 코어(14, 24)는 노드 내의 메모리(11, 21) 및 데이터 전송 수단들(13, 23)에 PCI Express 등을 통해서 액세스할 수 있다. 또한, 본 예시적인 실시예에서 정보 처리 장치는, NUMA(Non-Uniformed Memory Access) 유형의 정보 처리 장치이고, 하나의 노드가 다른 노드의 메모리에 액세스할 때(예를 들어, 노드(1)가 노드(2)의 메모리(21)에 액세스할 때), 노드간 통신 수단(3)이 사용된다.

- [0036] 메모리(11, 21)는 노드 내의 프로세서 코어들(14, 24)에 의해 공유된다.
- [0037] 데이터 전송 수단들(13, 23)은 CPU(12, 22)를 통하지 않고 메모리(11, 21)에 액세스하는 DMA(Direct Memory Access) 엔진(15, 25), 및 CPU(12, 22)에서의 레지스터와 같은 자원에 액세스하는 투-CPU(to-CPU) 통신 기능(16, 26)을 갖는다. 한편, 투-CPU 통신 기능(16, 26)은, CPU(12, 22)에서의 레지스터와 같은 자원을 메모리 공간으로 매핑하고 DMA 엔진(15, 25)을 통해 자원에 액세스하도록 구성될 수 있다.
- [0038] 노드간 통신 수단(3)은 PCI 익스프레스, 인피니밴드와 같은 DMA 기능 또는 RDMA(Remote Direct Memory Access) 기능을 갖는 인터커넥트에 의해 실현될 수 있다.
- [0039] 본 예시적인 실시예에서는, NUMA 유형 메모리들을 갖는 정보 처리 장치의 노드들이 다음의 방식으로 구별된다; 노드들 중 하나는 제어 노드로서 지칭되고, 다른 하나는 계산 노드로서 지칭된다. 본 예시적인 실시예에서는, 제어 노드가 운영 체제(OS) 기능을 갖는 반면, 계산 노드는 운영 체제 기능을 갖지 않는다. 또한, 계산 노드는 정확한 예외를 지원하지 않는 CPU(22)를 갖는다. 또한, CPU(22)는 비순차적 실행을 행하도록 구성될 수 있거나, 순차적 실행을 행하도록 구성될 수 있다. 도 1에서는, 노드(1)가 제어 노드이고, 노드(2)가 계산 노드이다. 즉, 제어 노드(1)는 메모리(11), CPU(12) 및 데이터 전송 수단(13)을 포함한다. 또한, 계산 노드(2)는 메모리(21), CPU(22) 및 데이터 전송 수단(23)을 포함한다. 그 후, 계산 노드(2)에서의 프로세서 코어(24)가 계산 코어로서 지칭된다.
- [0040] 도 2를 참조하면, 계산 코어(24)는 범용 레지스터 세트(240), 제어 레지스터 세트(241), 예외 검출 수단(242), 레지스터 세트 액세스 수단(243) 및 예외 통지 수단(244)을 포함한다.
- [0041] 범용 레지스터 세트(240)는 x86 및 ARM과 같은 일반적인 프로세서와 마찬가지로, 프로그램 카운터(PC), 계산 결과를 저장하기 위한 범용 레지스터(GPR) 등을 포함한다. 이들 레지스터들은 계산 코어(24)에서 명령어의 실행에 의해 재기입된다. 또한, 계산 코어(24)가 로드 명령어 또는 저장 명령어를 실행하는 경우, 범용 레지스터 세트(240)와 메모리 사이에서 데이터가 전송된다.
- [0042] 제어 레지스터 세트(241)는 계산 코어(24)의 실행을 제어하기 위한 레지스터이다. 제어 노드(1)는 계산 노드(2)의 제어 레지스터(241)에게 노드간 통신 수단(3) 및 데이터 전송 수단들(13, 23)을 통해 명령어의 실행을 개시하고 정지하도록 지시한다. 제어 노드(1)가 계산 코어(24)에게 실행을 개시하도록 지시하는 경우, 계산 코어(24)는 범용 레지스터(240)의 콘텐츠에 따라 처리를 개시한다. 구체적으로, 계산 코어(24)는 프로그램 카운터의 값을 따라 명령어를 페치하고, 페치된 명령어를 실행하고, 범용 레지스터 세트(240) 및 메모리(21)의 콘텐츠를 갱신한다. 이 상태는 명령어 실행 상태로서 지칭된다. 한편, 제어 노드(1)가 계산 코어(24)에게 실행을 정지하도록 지시하는 경우, 계산 코어(24)가 신규 명령어의 실행을 정지시킨다. 실행을 정지하라는 지시가 주어지고, 계산 코어(24)에 진행 중인 명령어가 없는 상태는 명령어 실행 정지 상태로서 지칭된다. 계산 코어(24)는 일단 명령어 실행 정지 상태로 진행하면, 제어 노드(1)에 의해 실행을 개시하도록 지시받을 때까지, 신규 명령어를 실행하지 않는다. 계산 코어(24)가 명령어 실행 상태에 있는지 또는 명령어 실행 정지 상태에 있는지가 노드간 통신 수단(3), 데이터 전송 수단(13) 및 레지스터 세트 액세스 수단(243)을 통해 제어 노드(1)에 의해 참조될 수 있다. 참조하기 위한 기능이 제공되지 않는 경우에, 예외 정보를 내보내고, 이로써 계산 코어가 명령어 실행 상태에 있는지 또는 명령어 실행 정지 상태에 있는지를 제어 노드에 통지하도록 구성할 수 있다.
- [0043] 예외 검출 수단(242)은, 명령어의 실행 동안 예외를 검출한 경우에, 제어 레지스터 세트(241)에 실행을 정지하도록 지시한다. 동시에, 예외 검출 수단(242)은 예외 통지 수단(244)에게 예외 통지를 요구한다(예외 발생 정보를 송신한다). 전송한 예외는 제로에 의한 분할 예외를 포함하는 연산 예외, 메모리 액세스 경계 오류를 포함하는 메모리 액세스 예외, 및 시스템 호출을 호출하기 위한 소프트웨어 트랩과 같은 일반적인 프로세서에서 발생하는 예외이다.
- [0044] 예외 통지 수단(244)은, 예외 검출 수단(242)으로부터 취득된 예외 발생 정보에 기초하여, 노드간 통신 수단(3)을 통해 제어 노드(1)에 계산 코어(24)에 의한 실행의 정지를 통지한다. 예외 통지 수단(244)은, 그것이 예

의 발생을 제어 노드(1)에 통지할 수 있는 한, 임의의 방법에 의해 구현될 수 있다. 따라서, 예외 통지 수단(244)은 다음의 방법들에 의해 실현될 수 있다; (A) PCI 익스프레스의 MSI/MSI-X와 같은 노드간 통신 수단(3)의 인터럽트 기능, (B) 사전에 예약한 메모리 영역으로의 DMA 기입. 방법 (B)가 사용되는 경우에, 제어 노드(1)에서의 후술하는 계산 노드 관리 수단(120)이, 계산 노드(2)로부터의 예외 통지의 존재 또는 부재를 폴링과 같은 방법에 의해 모니터링할 필요가 있다. 한편, 예외 통지 수단(244)은, 제어 노드(1)에 의한 설정에 따라, 제어 노드(1)로의 예외 발생 통지를 억제할 수 있다. 이 경우에, (C) 제어 노드(1)가 직접 제어 레지스터 세트(241)의 실행 상태를 모니터링(폴링)하고, 이로써 예외의 발생을 검출하도록 구성할 수 있다.

[0045] 레지스터 세트 액세스 수단(243)은, 제어 노드(1)에 의해 발행된 명령어에 기초하여, 범용 레지스터 세트(240) 및 제어 레지스터 세트(241)의 각각의 레지스터로부터의 판독 및 그에 대해 기입을 행한다. 범용 레지스터 세트(240)로부터의 판독 및 그에 대한 기입의 발행은 명령어 실행 정지 상태의 계산 코어(24)에서만 허용된다. 명령어 실행 상태의 계산 코어(24)의 범용 레지스터 세트(240)에 액세스가 있는 경우, 계산 코어(24)의 동작은 불안정하다(예를 들어, 액세스가 무시된다).

[0046] 본 예시적인 실시예에서 정보 처리 장치는 예를 들어 전술한 바와 같은 구성을 갖는다. 즉, 본 예시적인 실시예의 정보 처리 장치는 하나의 제어 노드(1) 및 하나 이상의 계산 노드(2)를 포함한다. 운영 체제가 제어 노드(1)에 통합되어 있다. 제어 노드(1)는 계산 노드(2)에서 복수의 프로세스를 전개하고, 동시에 실행된다. 한편, 운영 체제가 계산 노드(2)에서 실행되지 않는다. 또한, 본 예시적인 실시예에서 계산 코어(24)는 정확한 예외를 지원하지 않는다.

[0047] 도 3에 도시된 바와 같이, 본 예시적인 실시예에서 정보 처리 장치는 제어 노드(1)에서의 메모리(11)(제1 메모리) 내의 데이터의 콘텐츠를 계산 노드(2)에서의 메모리(21)(제2 메모리) 내의 데이터의 콘텐츠와 동기화하여 제어 노드(1)와 계산 노드(2) 사이에 실질적인 메모리 공유를 허용한다. 구체적으로, 후술하는 바와 같이, 본 예시적인 실시예에서 정보 처리 장치는, 메모리 매핑된 파일의 공유 매핑에 의해 공유 메모리들을 제어 노드(1) 및 계산 노드(2)에서 생성하고, 제어 노드(1) 및 계산 노드(2)에 생성된 공유 메모리들을 적절한 타이밍에 갱신하고, 이로써 실질적인 메모리 공유를 허용한다.

[0048] 도 4는, 본 예시적인 실시예에서 정보 처리 장치에서, 전술한 바와 같은 실질적인 메모리 공유를 허용하기 위한 구성의 일례를 도시한다. 이하, 계산 노드(2) 및 제어 노드(1)가 더 상세하게 설명될 것이다.

[0049] <계산 노드>

[0050] 전술한 바와 같이, 운영 체제가 계산 노드(2)에서 실행되지 않는다. 따라서, OS 지터와 같은 OS의 악영향이 제거될 수 있다. 또한, 계산 노드(2)의 계산 코어(24)는 정확한 예외를 지원하지 않는다. 예를 들어 계산 노드(2)에서 더 높은 성능을 얻는 것이 요구되는 경우에 이 구성이 고려된다.

[0051] 이 계산 노드(2)에 일반적인 컴퓨터에 의해 실현되는 계산 모델을 사용하기 위해, 제어 노드(1)가 일반적인 운영 체제에 의해 제공되는 서비스의 대체 기능을 노드간 통신 수단(3)을 개재해서 계산 노드(2)에 제공한다.

[0052] 도 4에서 도시된 바와 같이, 복수의 계산 프로세스(200-m)가 계산 노드(2)에서 전개된다. 본 명세서에서, 계산 프로세스(200-m)는, 계산 노드(2)에서 전개된 프로세스들이다. 계산 프로세스(200-m)는 예를 들어, 계산 노드(2)에서 실행되는 애플리케이션 프로그램의 개시에 따라 생성되고, 예를 들어 그 실행의 종료에 따라 파기된다.

[0053] 구체적으로, 도 5에서 도시된 바와 같이, 계산 프로세스(200-m)에 관한 데이터인 복수의 프로세스 이미지(210-m)가 계산 노드(2)의 메모리(21) 내에 저장된다. 또한, 하나의 계산 프로세스(200-m)의 컨텍스트가 각각의 계산 코어(24-m) 내에 저장된다. 각각의 계산 코어(24-m)는 저장된 컨텍스트에 대응하는 메모리(21)에서의 프로세스 이미지(210-m)를 사용하는 것에 의해 계산 프로세스(200-m)의 처리를 행한다. 본 명세서에서, 메모리(21)에 저장되는 프로세스 이미지(210-m)의 수는 계산 코어(24-m)들의 수와 상이할 수 있다.

[0054] 또한, 공유 메모리(211)는 후술하는 프로세스에 의해 계산 노드(2)의 메모리(21) 내에 생성된다(도 4 참조). 공유 메모리(211)에는, 계산 프로세스(200-m)에 의해 취득된 공유 메모리가 배열된다. 후술하는 바와 같이, 본 예시적인 실시예에서 정보 처리 장치는 공유 메모리(211)의 데이터의 콘텐츠가 제어 노드(1)의 공유 메모리(111)의 데이터의 콘텐츠와 동기화된다.

[0055] 또한, 계산 노드(2)는, 계산 프로세스(200-m)가 공유 메모리(211) 내의 데이터를 갱신하는 경우 공유 메모리(211) 내의 데이터가 변경된 것을 나타내는 공유 메모리 갱신 식별자(201)를 갖는다. 계산 프로세스(200-m)가 공유 메모리(211)를 갱신하는 경우, 공유 메모리(211) 내의 데이터가 갱신된 것을 나타내도록, 공유 메모리 갱

신 식별자(201)는 계산 프로세스(200-m)에 의해 조작된다.

- [0056] <제어 노드>
- [0057] 제어 노드(1)는 운영 체제를 사용하는 일반적인 독립형의 컴퓨터 시스템과 동일한 구성을 갖고, 계산 노드(2)를 위한 운영 체제 기능의 서비스를 제공한다. 본 명세서에서, 제어 노드(1)에 통합되는 운영 체제는 제어 노드 OS(110)로서 지칭될 것이다. 제어 노드 OS(110)는 예를 들어 메모리 매핑된 파일 기능과 같은 운영 체제로서 일반적인 기능을 갖는다. 제어 노드 OS(110)는, Linux(등록 상표) 및 Windows(등록 상표)와 같이 컴퓨터에 사용되는 상용 운영 체제일 수 있다.
- [0058] 계산 노드(2)와 달리, 제어 노드(1)는 제어 노드 OS(110)가 실행되기 때문에 제어 노드 OS(110)의 시스템 관리 기능을 사용하는 것에 의해, 하드웨어(프로세서 코어(14) 및 메모리(11))를 가상화할 수 있다. 또한, 계산 노드(2)에서와 동일한 방식으로, 후술하는 프로세스를 통해, 공유 메모리(111)가 제어 노드(1)의 메모리(11)에 생성된다.
- [0059] 제어 노드(1)는 제어 노드 OS(110) 상에 복수의 대체 프로세스(120-m) 및 계산 노드 관리 수단(130)을 갖는다. 또한, 제어 노드 OS(110)는 예를 들어, 페이지 테이블(1101), 메모리 속성 설정 수단(1103) 및 시스템 호출(1104)의 기능들을 갖는다.
- [0060] 페이지 테이블(1101)은 페이징을 사용하는 경우 메모리에 관한 정보를 유지한다. 페이지 테이블(1101)에서, 가상 어드레스와 물리 어드레스의 매핑이 저장된다. 또한, 페이지 테이블(1101)은 메모리의 하나의 페이지에 대응하는 페이지 테이블 엔트리(1102)를 갖는다. 페이지의 사용 상태 및 페이지에 설정되는 속성은 페이지 테이블 엔트리(1102)에 유지된다.
- [0061] 메모리 속성 설정 수단(1103)은 후술하는 공유 메모리(111)의 취득 시에 메모리의 속성들을 설정한다. 예를 들어, 메모리 속성 설정 수단(1103)은 공유 메모리(111)의 취득 시에 메모리의 속성을 리드-온리(read-only)로 설정한다.
- [0062] 시스템 호출(1104)은 제어 노드(1) 및 계산 노드(2)에 OS 기능을 제공한다. 예를 들어, 계산 노드(2)는 대체 프로세스(120-m)를 개재해서 시스템 호출(1104)을 이용한다.
- [0063] 대체 프로세스(120-m)들은 계산 노드(2)에서의 각자의 계산 프로세스(200-m)들에 대응하도록 생성된다. 따라서, 대체 프로세스(120-m)들 중 하나가 계산 노드(2)에서의 프로세스 이미지(210-m)들 중 하나에 대응한다. 대체 프로세스(120-m)는 계산 노드(2)에서의 계산 프로세스(200-m)와 연관되고, 계산 프로세스(200-m)에 의해 발행되는 시스템 호출 요청을 노드간 통신 수단(3) 및 계산 노드 관리 수단(130)을 개재해서 수락한다. 대체 프로세스(120-m)는 시스템 호출(1104)을 사용하는 것에 의해 계산 노드(2)에 존재해야 할 운영 체제 대신에 시스템 호출 처리를 행한다.
- [0064] 대체 프로세스(120-m)는 시스템 호출 요청을 대체 실행하기 위한 복수의 기능을 갖는다. 예를 들어, 대체 프로세스(120-m)는 메모리 갱신 대체 실행 수단들(1201-m) 및 메모리 매핑 대체 실행 수단들(1202-m)로서의 기능들을 갖는다. 메모리 갱신 대체 실행 수단들(1201-m)은 계산 프로세스(200-m)의 메모리 갱신을 행하는 시스템 호출을 대체 실행한다. 메모리 매핑 대체 실행 수단들(1202-m)은 계산 프로세스(200-m)에 의한, 본 발명에 따른 메모리 매핑된 파일 기능의 실행을 위해 사용된다.
- [0065] 또한, 대체 프로세스(120-m)는 대체 프로세스(120-m)에 의해 사용되는 공유 메모리(111)와 계산 프로세스(200-m)에 의해 사용되는 공유 메모리(211)의 대응 상태를 유지하는 공유 메모리 정보(1203-m)를 갖는다. 예를 들어, 공유 메모리 정보(1203-m)에는, 대체 프로세스(120-m)에 의해 사용되는 공유 메모리(111)의 영역 정보 및 대응하는 계산 프로세스(200-m)에 의해 사용되는 공유 메모리(211)의 영역 정보가 서로 연관되도록 저장된다.
- [0066] 계산 노드 관리 수단(130)은 계산 노드(2)에 대하여 생성되고, 계산 노드(2)에서의 프로세서 코어(24)(계산 코어(24)) 및 메모리(21)와 같은 하드웨어 자원들을 관리한다. 따라서, 정보 처리 장치가 복수의 계산 노드(2)를 갖는 경우에, 계산 노드(2)들 각각에 대해 계산 노드 관리 수단(130)이 생성된다. 또한, 계산 노드 관리 수단(130)은 일부의 자원에 대한 대체 프로세스(120-m)에 의해 계산 코어(24)에 액세스하기 위한 중개자로서 역할을 한다. 예를 들어, 계산 노드 관리 수단(130)은 계산 프로세스(200-m)를 생성할 때 계산 코어(24)를 할당하고, 계산 노드(2)에서의 메모리를 관리하고, 계산 프로세스(200-m)로부터 대체 프로세스(120-m)로의 시스템 호출을 위한 중개자로서 역할을 한다.
- [0067] 전술한 동작을 실현하기 위해서, 계산 노드 관리 수단(130)은 계산 노드(2)에서의 계산 코어(24) 및 메모리(2

1)와 같은 하드웨어 자원들을, 계산 프로세스(200-m)들 및 대체 프로세스(120-m)들과 같은 가상화된 자원들에 링크하고, 그것들을 관리한다. 예를 들어, 계산 노드 관리 수단(130)은 대체 프로세스(120-m)들과 계산 코어(24)들의 대응을 프로세스 대응 테이블(1303)에 저장한다.

[0068] 도 6은 프로세스 대응 테이블(1303)의 일례를 도시한다. 도 6을 참조하면, 프로세스 대응 테이블(1303)에서는, 대체 프로세스 ID 13031과 계산 코어 번호 13032가 서로 링크되어 있다. 대체 프로세스 ID 13031은 대체 프로세스(120-m)를 식별하기 위한 ID이다. 계산 코어 번호 13032는 대체 프로세스(120-m)에 대응하는 계산 프로세스(200-m)가 실행되고 있는 계산 코어(24)의 번호를 나타낸다. 예를 들어, 도 6의 제1행에서, 대체 프로세스(120-m) "31996"에 대응하는 계산 프로세스(200-m)가 실행되고 있는 계산 코어(24)의 번호가 "Core#0"인 것을 나타낸다.

[0069] 이러한 프로세스 대응 테이블(1303)의 사용은 계산 노드 관리 수단(130)이 계산 코어 번호 13032로부터, 대응하는 대체 프로세스 ID 13031을 얻을 수 있게 한다. 또한, 계산 노드 관리 수단(130)은 대체 프로세스 ID 13031로부터, 동작하고 있는 계산 코어(24)의 번호인 계산 코어 번호 13032를 얻을 수 있다.

[0070] 또한, 계산 노드 관리 수단(130)은 예를 들어, 계산 노드 메모리 갱신 확인 수단(1301) 및 공유 메모리 동기화 수단(1302)으로서의 기능들을 갖는다. 계산 노드 메모리 갱신 확인 수단(1301)은 계산 노드(2)에서의 공유 메모리(211) 내의 데이터 갱신 상태를 확인한다. 계산 노드 메모리 갱신 확인 수단(1301)은 예를 들어, 계산 노드(2)에서의 공유 메모리 갱신 식별자(201)를 참조하고, 이로써 공유 메모리(211) 내의 데이터의 콘텐츠가 갱신되는지를 확인한다. 공유 메모리 동기화 수단(1302)은 제어 노드(1)에서의 공유 메모리(111)를 계산 노드(2)에서의 공유 메모리(211)와 동기화한다. 예를 들어, 공유 메모리 동기화 수단(1302)은 계산 노드(2)에서의 공유 메모리(211) 내의 데이터의 갱신 콘텐츠를 제어 노드(1)에서의 공유 메모리(111)에 복사한다. 대신에, 예를 들어, 공유 메모리 동기화 수단(1302)은 제어 노드(1)에서의 공유 메모리(111) 내의 데이터의 갱신 콘텐츠를 계산 노드(2)에서의 공유 메모리(211)에 복사한다.

[0071] 이상은 본 예시적인 실시예에서 정보 처리 장치에 의해 포함된 계산 노드(2) 및 제어 노드(1)에 대한 상세한 설명이다. 계산 노드(2) 및 제어 노드(1)는 전술한 구성을 사용하는 것에 의해, 공유 메모리들(111 및 211)을 취득한다. 또한, 계산 노드(2) 및 제어 노드(1)는 후술하는 미리 결정된 타이밍에 공유 메모리들(111 및 211)을 동기화한다. 이하, 공유 메모리들(111 및 211)을 취득할 때 및 공유 메모리들(111 및 211)을 동기화할 때 본 예시적인 실시예에서 정보 처리 장치의 동작이 설명될 것이다.

[0072] 먼저, 계산 노드(2)에서의 계산 프로세스(200-m)가 메모리 매핑된 파일의 공유 매핑에 의한 공유 메모리들(111 및 211)을 취득할 때 동작의 일례가 도 7 및 도 9를 사용하는 것에 의해 설명될 것이다.

[0073] [공유 메모리 취득 시]

[0074] 도 7은 계산 노드(2)에 의한 처리를 나타내는 흐름도이다. 도 7을 참조하면, 계산 노드(2)에서 실행되는 계산 프로세스(200-m)는 제어 노드(1)의 대체 프로세스(120-m)에 공유 메모리들(111 및 211)을 할당(시스템 호출을 실행)하도록 요청한다(단계 S101). 계산 프로세스(200-m)는 할당을 요청하는 것과 함께, 자체의 실행을 정지한다(단계 S102). 그 결과, 계산 프로세스(200-m)는 대체 프로세스(120-m)가 시스템 호출을 대체 실행하고 계산 프로세스(200-m)의 실행을 재개시킬 때까지 대기한다(계산 프로세스(200-m)는 대체 프로세스(120-m)에 의해 재개될 때까지 정지한다). 공유 메모리들(111 및 211)의 할당이 대체 프로세스(120-m)에 의해 완료된 후, 대체 프로세스(120-m)에 의해, 계산 프로세스(200-m)의 실행이 재개된다(단계 S103).

[0075] 도 8은 제어 노드(1)에 의한 처리를 나타내는 흐름도이다. 도 8을 참조하면, 계산 노드 관리 수단(130)은 계산 노드(2)(계산 프로세스(200-m))에 의해 발행되는 공유 메모리 할당의 시스템 호출을 취득한다(단계 S111). 그 후, 계산 노드 관리 수단(130)은 취득된 시스템 호출을 대체 프로세스(120-m)에 통지한다. 따라서, 공유 메모리 할당의 시스템 호출이 대체 프로세스(120-m)에 의해 개시된다.

[0076] 제어 노드 OS(110)에 의해 제공되는 메모리 매핑된 파일 기능에 의해, 대체 프로세스(120-m)는 메모리 매핑 대체 실행 수단들(1202-m)로 하여금 외부 디스크 등으로부터 공유될 파일을 검색하고, 공유 메모리(111)를 제어 노드(1)의 메모리(11)에 생성하게 한다(단계 S112). 또한, 대체 프로세스(120-m)의 메모리 매핑 대체 실행 수단들(1202-m)은 메모리 속성 설정 수단(1103)으로 하여금 생성된 공유 메모리(111)의 속성을 리드-온리로 설정하게 한다(단계 S113).

[0077] 후속하여, 대체 프로세스(120-m)의 메모리 매핑 대체 실행 수단들(1202-m)은 계산 노드(2)의 메모리(21)에 공유 메모리(211)의 영역을 생성한다. 그 후, 메모리 매핑 대체 실행 수단들(1202-m)은 제어 노드(1)에서의 공유 메

모리(111)의 콘텐츠를 계산 노드(2)에서의 공유 메모리(211)에 복사한다(단계 S114).

- [0078] 그 후, 대체 프로세스(120-m)는 제어 노드(1)에서의 생성된 공유 메모리(111) 및 제어 노드(2)에서의 대응하는 공유 메모리(211)의 영역 정보를 공유 메모리 정보(1203-m)에 유지한다(단계 S115). 그 후, 대체 프로세스(120-m)는 계산 프로세스(200-m)의 실행을 재개시한다(단계 S116).
- [0079] 이상이고, 공유 메모리(111 및 211)를 취득할 때의 동작의 일례이다. 다음으로, 제어 노드(1)에 생성된 공유 메모리(111)를 계산 노드(2)에 생성된 공유 메모리(211)와 동기화할 때의 동작이 설명될 것이다.
- [0080] 본 예시적인 실시예에서 정보 처리 장치는 계산 프로세스(200-m)가 실행을 정지하고 있는 동안, 공유 메모리(111)를 공유 메모리(211)와 동기화한다. 구체적으로는, 정보 처리 장치는 이하에 나타낸 4개의 타이밍에서 공유 메모리(111)를 공유 메모리(211)와 동기화한다;
- [0081] (1) 계산 노드(2)의 계산 프로세스(200-m)가 메모리 갱신(공유 메모리(211)의 갱신)의 시스템 호출을 발행할 때,
- [0082] (2) 계산 프로세스(200-m)가 명령어 실행 정지 상태로 천이할 때(계산 프로세스(200-m)가 CPU 명령어에 의해 공유 메모리(211)를 갱신하는 경우),
- [0083] (3) 제어 노드(1)가 시스템 호출에 따라 공유 메모리(111)를 갱신하고, 그 후 대응하는 계산 프로세스(200-m)가 명령어 실행 정지 상태로 천이했을 때, 및
- [0084] (4) 제어 노드(1)가 CPU 메모리에 따라 공유 메모리(111)를 갱신하고, 그 후 대응하는 계산 프로세스(200-m)가 명령어 실행 정지 상태로 천이했을 때
- [0085] [타이밍 (1)]
- [0086] 먼저, 계산 노드(2)의 계산 프로세스(200-m)가 시스템 호출을 발행하여 메모리 갱신을 행할 때 실행되는 동기화 처리의 일례가 도 9 및 도 10을 사용하는 것에 의해 설명될 것이다. 도 9는 계산 노드(2)에 의해 실행되는 처리를 나타내는 흐름도이다. 도 9를 참조하면, 계산 노드(2)에서 실행되는 계산 프로세스(200-m)는 제어 노드(1)의 대체 프로세스(120-m)에 대해 메모리 갱신을 행하는 시스템 호출을 발행한다(단계 S201). 계산 프로세스(200-m)는 시스템 호출을 발행함과 함께, 자체의 실행을 정지한다(단계 S202). 따라서, 계산 프로세스(200-m)는 대체 프로세스(120-m)가 시스템 호출을 대체 실행하고 계산 프로세스(200-m)의 실행을 재개시킬 때까지 대기한다(계산 프로세스(200-m)는 대체 프로세스(120-m)에 의해 재개될 때까지 정지한다). 시스템 호출의 실행 및 메모리들의 동기화가 대체 프로세스(120-m)에 의해 행해진 후, 대체 프로세스(120-m)에 의해, 계산 프로세스(200-m)의 실행이 재개시된다(단계 S203).
- [0087] 도 10은 제어 노드(1)에 의해 실행되는 처리를 나타내는 흐름도이다. 도 10을 참조하면, 계산 노드 관리 수단(130)은 계산 노드(2)(계산 프로세스(200-m))에 의해 발행되는 메모리 갱신을 행하는 시스템 호출을 취득한다(단계 S211). 그 후, 계산 노드 관리 수단(130)은 취득된 시스템 호출을 대체 프로세스(120-m)에 통지한다. 따라서, 메모리 갱신을 행하는 시스템 호출 처리가 대체 프로세스(120-m)에 의해 개시된다.
- [0088] 대체 프로세스(120-m)는 메모리 갱신 대체 실행 수단들(1201-m)을 사용하여, 시스템 호출 요청에 기초하여 계산 노드(1)에서의 메모리(211)의 콘텐츠를 갱신한다(단계 S212). 후속하여, 대체 프로세스(120-m)의 메모리 갱신 대체 실행 수단들(1201-m)은 공유 메모리 정보(1203-m)를 참조하고 데이터의 상기의 갱신이 공유 메모리(211) 내의 데이터의 갱신인지 여부를 확인한다(단계 S213).
- [0089] 상기의 시스템 호출에 기초한 데이터의 갱신이 공유 메모리(211)의 데이터의 갱신인 경우(단계 S213, 예), 대체 프로세스(120-m)는 공유 메모리 동기화 수단(1302)을 사용하여, 공유 메모리(211)의 갱신 콘텐츠를 제어 노드(1)의 공유 메모리(111)에 복사한다(단계 S214). 그 후, 대체 프로세스(120-m)는 계산 프로세스(200-m)의 실행을 재개시한다(단계 S215).
- [0090] 한편, 시스템 호출에 기초한 데이터의 갱신이 공유 메모리(211) 내의 데이터의 갱신이 아닌 경우(단계 S213, 아니오), 대체 프로세스(120-m)는 공유 메모리(111)를 공유 메모리(211)와 동기화하지 않고 계산 프로세스(200-m)의 실행을 재개시한다(단계 S215).
- [0091] 본 예시적인 실시예에서 정보 처리 장치는 일례로서 전술한 바와 같은 처리를 통해, 공유 메모리(111)를 공유 메모리(211)와 동기화한다. 즉, 도 11을 참조하면, 계산 노드(2)의 계산 프로세스(200)가 제어 노드(1)의 대체 프로세스(120)에게, 공유 메모리(211)를 갱신하기 위해 시스템 호출을 대체 실행하도록 요청한다(Y1). 상기 요

청과 함께, 계산 프로세스(200)는 정지한다. 후속하여, 요청을 수신한 대체 프로세스(120)는 시스템 호출 요청에 따라, 계산 노드(2)에서의 공유 메모리(211)의 콘텐츠를 갱신한다(Y2). 그 후, 대체 프로세스(120)는 공유 메모리(211) 내의 데이터의 갱신 콘텐츠를 공유 메모리(111)에 복사한다(Y3). 그 후, 대체 프로세스(120)는 정지해 있던 계산 프로세스(200)를 재개시한다.

[0092] [타이밍 (2)]

[0093] 다음으로, 계산 프로세스(200-m)가 명령어 실행 정지 상태로 천이할 때 실행되는 동기화 처리의 일례가 도 12를 사용하는 것에 의해 설명될 것이다.

[0094] 전술한 바와 같이, 계산 노드(2)에서는, 운영 체제가 실행되지 않아서 타이머 인터럽트와 같은 정기적인 처리가 수행되지 않는다. 따라서, 프로세스 스케줄링 등을 위해서, 제어 노드(1)는 정기적으로 계산 프로세스(200-m)로 하여금 명령어 실행 상태에서부터 명령어 실행 정지 상태로 천이하게 한다. 따라서, 제어 노드(1)는 계산 프로세스(200-m)가 명령어 실행 정지 상태로 천이한 타이밍에서, 계산 프로세스(200-m)에 의해 갱신된 공유 메모리(211)의 콘텐츠를 공유 메모리(111)에 복사한다. 따라서, 공유 메모리(111) 및 공유 메모리(211)가 동기화된다. 또한, 계산 프로세스(200-m)는 처리를 실행하는 동안, CPU 명령어에 따라 공유 메모리(211) 내의 데이터를 갱신할 수 있다. 이 경우, 본 예시적인 실시예에서 계산 프로세스(200-m)는 공유 메모리 갱신 식별자(201)를 사용하여, 공유 메모리(211) 내의 데이터를 갱신되어 있는 것을 나타낸다.

[0095] 도 12를 참조하면, 대체 프로세스(120-m)는 계산 프로세스(200-m)가 프로세스 스위치 등에 의해 실행을 정지한 타이밍에서, 계산 노드 메모리 갱신 확인 수단(1301)을 사용하여 공유 메모리 갱신 식별자(201)를 확인한다(단계 S311).

[0096] 공유 메모리(211) 내의 데이터가 갱신되는 경우(단계 S312, 예), 대체 프로세스(120-m)는 공유 메모리 동기화 수단(1302)을 사용하여 공유 메모리(211)의 갱신 콘텐츠를 제어 노드(1)의 공유 메모리(111)에 복사한다(단계 S313). 한편, 공유 메모리(211) 내의 데이터가 갱신되지 않은 경우(단계 S312, 아니오), 대체 프로세스(120-m)는 공유 메모리(111)를 공유 메모리(211)와 동기화하지 않는다. 그 후, 프로세스 스위치와 같은 다른 처리가 계속된다.

[0097] 일례로서 전술한 바와 같은 처리를 통해, 정보 처리 장치는 계산 프로세스(200-m)가 명령어 실행 정지 상태로 천이할 때 공유 메모리(111)를 공유 메모리(211)와 동기화한다. 즉, 도 13을 참조하면, 계산 프로세스(200)가 CPU 명령어에 의해 공유 메모리(211) 내의 데이터를 갱신한다(Y11). 그 후, 계산 프로세스(200)가 프로세스 스위치 등에 의해 실행을 정지한다. 그 후, 계산 프로세스(200)가 정지한 타이밍에서, 대체 프로세스(120)는 계산 노드 메모리 갱신 확인 수단(1301)을 사용하여 공유 메모리 갱신 식별자(201)를 확인한다(Y12). 공유 메모리(211) 내의 데이터가 갱신되어 있는 경우, 대체 프로세스(120)는 공유 메모리 동기화 수단(1302)을 사용하여 공유 메모리(211) 내의 데이터의 갱신 콘텐츠를 공유 메모리(111)에 복사한다(Y13). 그 후, 프로세스 스위치와 같은 다른 처리가 계속된다.

[0098] 전술한 바와 같이, 계산 코어(24)가 명령어 실행 상태에 있는지 또는 명령어 실행 정지 상태에 있는지가 제어 노드(1)에 의해 참조될 수 있다. 따라서, 대체 프로세스(120-m)는 계산 프로세스(200-m)가 실행 상태에 있는지 또는 실행 정지 상태에 있는지를 결정하는 것이 가능하다.

[0099] [타이밍 (3)]

[0100] 다음으로, 제어 노드(1)에서의 프로세스가 시스템 호출에 의해 공유 메모리(111)를 갱신할 때, 공유 메모리(111)의 공유 메모리(211)와의 동기화의 일례가 도 14를 사용하는 것에 의해 설명될 것이다.

[0101] 도 14를 참조하면, 제어 노드(1)에서의 프로세스가 시스템 호출(1104)을 사용하여, 공유 메모리(111)를 갱신하기 위해 시스템 호출을 실행한다(단계 S411). 따라서, 공유 메모리(111)의 콘텐츠가 갱신된다(단계 S412).

[0102] 그 후, 제어 노드(1)는 공유 메모리 정보(1203-m)를 참조하여, 공유 메모리(111)의 갱신된 영역에 대응하는 공유 메모리(211)의 영역을 특정한다. 그 후, 제어 노드(1)는 전술된 공유 메모리(211)의 영역을 사용하는 계산 프로세스(200-m)의 명령어 실행 상태를 확인한다(단계 S413). 후속하여, 계산 프로세스(200-m)가 실행을 정지한 타이밍에서(단계 S413, 예), 제어 노드(1)는 공유 메모리 동기화 수단(1302)으로 하여금 공유 메모리(111)의 갱신 콘텐츠를 공유 메모리(211)에 복사하게 한다(단계 S414). 따라서, 공유 메모리(111)와 공유 메모리(211)는 서로 동기화된다. 이 때, 계산 노드(2)에서의 공유 메모리(211)가 의도하지 않는 타이밍에서 갱신되는 것을 방지하기 위해서, 배타적 제어가 적절히 실행된다.

- [0103] 일례로서, 전술한 바와 같은 처리를 통해, 정보 처리 장치는 제어 노드(1)에서의 프로세스가 시스템 호출에 의해 공유 메모리(111)를 갱신한 경우에, 공유 메모리(111)의 공유 메모리(211)와의 동기화를 행한다. 즉, 도 15를 참조하면, 제어 노드(1)에서의 프로세스가 시스템 호출(1104)을 사용하여, 공유 메모리(111)를 갱신하기 위해 시스템 호출을 실행한다(y1). 그 후, 제어 노드(1)는 공유 메모리 정보(1203-m)를 참조하여, 공유 메모리(111)의 갱신된 영역에 대응하는 공유 메모리(211)의 영역을 특정한다. 그 후, 제어 노드(1)는 전술한 바와 같이 공유 메모리(211)의 영역을 사용하는 계산 프로세스(200-m)의 명령어 실행 상태를 확인한다. 그 후, 특정된 공유 메모리(211)의 영역을 사용하는 계산 프로세스(200-m)가 실행을 정지한 타이밍에서, 제어 노드(1)는 공유 메모리 동기화 수단(1302)을 사용하여, 공유 메모리(111) 내의 데이터의 갱신 콘텐츠를 공유 메모리(211)에 복사한다(y2).
- [0104] [타이밍 (4)]
- [0105] 다음으로, 제어 노드(1)에서의 프로세스가 CPU 명령어에 의해 공유 메모리(111)를 갱신할 때, 공유 메모리(111)의 공유 메모리(211)와의 동기화의 일례가 도 16을 사용하는 것에 의해 설명될 것이다.
- [0106] 도 16을 참조하면, 제어 노드(1)에서의 프로세스가 CPU 명령어에 의해 공유 메모리(111)를 갱신하려고 한다(단계 S511). 전술한 바와 같이, 타깃 영역에 대해 메모리 속성 설정 수단(1103)에 의해 리드-온리의 속성이 부여되어 있기 때문에, 리드-온리 예외가 발생한다(단계 S512). 따라서, 제어 노드(1)는 상기의 예외 핸들링을 행하고(단계 S513), 예외 핸들링 동안, 리드-온리를 일시 해제하고 공유 메모리(111)를 갱신한다(단계 S514). 공유 메모리(111)를 갱신한 후, 제어 노드(1)는 메모리 속성 설정 수단(1103)을 사용하여, 공유 메모리(111)의 속성을 다시 리드-온리로 설정한다(단계 S515).
- [0107] 그 후, 제어 노드(1)는 공유 메모리 정보(1203-m)를 참조하여, 갱신된 공유 메모리(111)에 대응하는 공유 메모리(211)를 사용하는 계산 프로세스(200-m)의 명령어 실행 상태를 확인한다(단계 S516). 계산 프로세스(200-m)가 실행을 정지한 타이밍에서(단계 S516, 예), 제어 노드(1)는 공유 메모리 동기화 수단(1302)으로 하여금 갱신된 공유 메모리(111)의 콘텐츠를 공유 메모리(211)에 복사하게 한다(단계 S517). 따라서, 공유 메모리(111)와 공유 메모리(211)는 서로 동기화된다. 이 때, 계산 노드(2)에서의 공유 메모리(211)가 의도하지 않는 타이밍에서 갱신되는 것을 방지하기 위해서, 상호 배제가 적절히 수행된다.
- [0108] 일례로서, 전술한 바와 같은 처리를 통해, 정보 처리 장치는 제어 노드(1)에서의 프로세스가 CPU 명령어에 의해 공유 메모리(111)를 갱신한 경우에, 공유 메모리(111)의 공유 메모리(211)와의 동기화를 행한다. 즉, 도 17을 참조하면, CPU 명령어에 의해 공유 메모리(111)를 갱신하려고 할 때 야기되는 예외를 핸들링하는 동안, 제어 노드(1)에서의 프로세스가 리드-온리를 일시 해제하고 공유 메모리(111)를 갱신한다(y11). 그 후, 제어 노드(1)는 공유 메모리(111)의 속성을 다시 리드-온리로 설정한다. 그 후, 제어 노드(1)는 공유 메모리 정보(1203-m)를 참조하고, 공유 메모리(111)의 갱신된 영역에 대응하는 공유 메모리(211)의 영역을 특정한다. 그 후, 제어 노드(1)는 전술한 바와 같이 공유 메모리(211)의 영역을 사용하는 계산 프로세스(200-m)의 명령어 실행 상태를 확인한다. 그 후, 공유 메모리(211)의 특정된 영역을 사용하는 계산 프로세스(200-m)가 실행을 정지한 타이밍에서, 제어 노드(1)는 공유 메모리 동기화 수단(1302)을 사용하여, 공유 메모리(111) 내의 데이터의 갱신 콘텐츠를 공유 메모리(211)에 복사한다(y12).
- [0109] 상기의 처리를 통해, 제어 노드(1)의 공유 메모리(111) 및 계산 노드(2)의 공유 메모리(211)가 최신 상태에서 서로 동기화된다. 한편, 공유 소스의 파일의 갱신은 종래 OS(제어 노드 OS(110))에 의해 지원되는 메모리 매핑된 파일 기능에 의해 수행된다.
- [0110] 전술한 바와 같이, 본 예시적인 실시예에서 정보 처리 장치는 운영 체제가 통합되어 있는 제어 노드(1), 및 운영 체제가 통합되어 있지 않고 정확한 예외를 지원하지 않는 계산 노드(2)를 갖는다. 공유 메모리(111)가 제어 노드(1)의 메모리(11)에 생성되고, 공유 메모리(211)가 계산 노드(2)의 메모리(21)에 생성된다. 계산 프로세스(200-m)가 실행을 정지하는 동안, 제어 노드(1)의 대체 프로세스(120-m)가 공유 메모리 동기화 수단(1302)을 사용하여, 공유 메모리(111)의 공유 메모리(211)와의 동기화를 행한다. 이러한 구성은 제어 노드(1)의 공유 메모리(111)를 계산 노드(2)의 공유 메모리(211)와 동기화할 수 있게 한다. 그 결과, 정확한 예외를 지원하지 않는 CPU(22)를 포함하는 계산 노드(2)에서 메모리 매핑된 파일 기능이 이용될 수 있고, 프로세스들 사이에서 메모리 공유가 허용된다.
- [0111] 또한, 본 예시적인 실시예에서 정보 처리 장치는, 전술한 4개의 타이밍에서 공유 메모리(111)의 공유 메모리(211)와의 동기화를 행하도록 구성된다. 이러한 구성은 최신 상태에서 공유 메모리(111)를 공유 메모리(211)와

동기화할 수 있게 한다.

- [0112] 본 예시적인 실시예에서, 정보 처리 장치가 2개의 노드, 노드(1) 및 노드(2)를 포함하는 경우가 설명되었다. 그러나, 전술한 바와 같이, 정보 처리 장치는 3개 이상의 노드를 포함할 수 있다. 즉, 정보 처리 장치는 1개의 제어 노드(1) 및 2개 이상의 계산 노드(2)를 포함할 수 있다. 이 경우, 공유 메모리 동기화 수단(1302)은 대응하는 공유 메모리(211)들 각각을 공유 메모리(111)와 동기화한다.
- [0113] 또한, 계산 노드(2)의 공유 메모리(211)의 갱신을 제어 노드(1)에 반영하는 방법은 본 예시적인 실시예에서 설명된 경우에 한정되지 않는다.
- [0114] 예를 들어, 제어 노드(1)에서의 공유 메모리(111)를 공유하는 프로세스들의 페이지 테이블(1101)에서, 공유 메모리(111)의 페이지 테이블 엔트리(1102)의 속성이 무효로 미리 설정된다. 제어 노드(1)에서의 프로세스가 공유 메모리(111)에 액세스하고자 할 때, 페이지 부재 예외가 발생한다. 그 후, 예외 핸들링 동안, 계산 노드(2)에서의 공유 메모리(211)의 콘텐츠가 제어 노드(1)에서의 공유 메모리(111)에 복사된다. 예를 들어, 이러한 방법에 의해, 계산 노드(2)의 공유 메모리(211)의 갱신을 제어 노드(1)에 반영하는 것이 고려될 수 있다. 이 방법을 사용하는 경우에, 공유 메모리(211)의 갱신이 시스템 호출에 의한 메모리 갱신인지 또는 CPU 명령어에 의한 메모리 갱신인지를 결정하는 것이 요구되지 않는다.
- [0115] 또한, 예를 들어 계산 프로세스(200-m)가 프로그램 상에서 명시적으로 공유 메모리(211)의 갱신 콘텐츠를 공유 메모리(111)에 복사하기 위한 명령어를 실행하는 것이 또한 고려될 수 있다. 이 방법을 사용하는 경우에, 공유 메모리(211)의 갱신이 시스템 호출에 의한 메모리 갱신인지 또는 CPU 명령어에 의한 메모리 갱신인지를 결정하는 것이 요구되지 않는다.
- [0116] 또한, 제어 노드(1)의 공유 메모리(111)의 갱신을 계산 노드(2)에 반영하는 방법이 본 예시적인 실시예에서 설명된 경우에 한정되지 않는다.
- [0117] 예를 들어, 계산 프로세스(200-m)가 실행을 정지하는 타이밍에서, 대체 프로세스(120-m)가 제어 노드(1)에서의 공유 메모리(111)의 갱신 상태를 조사하도록 정보 처리 장치가 구성된다. 공유 메모리(111)의 갱신을 확인하는 것에 대한 일례로서, 페이지 테이블 엔트리(1102)의 더티(Dirty) 비트를 이용하는 것이 고려될 수 있다. 그 후, 공유 메모리(111)가 갱신되어 있는 경우, 제어 노드(1)는 공유 메모리 동기화 수단(1302)을 사용하여, 공유 메모리(111)의 갱신 콘텐츠를 공유 메모리(211)에 복사한다. 이 방법을 사용하는 경우에, 공유 메모리(211)의 갱신이 시스템 호출에 의한 메모리 갱신인지 또는 CPU 명령어에 의한 메모리 갱신인지를 결정하는 것이 요구되지 않는다.
- [0118] [제2 예시적인 실시예]
- [0119] 다음으로, 본 발명의 제2 예시적인 실시예가 도 18을 참조하여 설명될 것이다. 본 예시적인 실시예에서, 메인 프로세서 코어와 서브 프로세서 코어를 갖고, 각자의 코어들에 생성된 공유 메모리들의 동기화를 행하는 정보 처리 장치의 개략적인 구성이 설명될 것이다.
- [0120] 도 18을 참조하면, 본 예시적인 실시예에서 정보 처리 장치는 운영 체제가 통합되어 있는 메인 프로세서 코어(4), 및 메인 프로세서 코어(4)에 의해 제어되고 운영 체제가 통합되어 있지 않은 서브 프로세서 코어(5)를 갖는다.
- [0121] 메인 프로세서 코어(4)는 제1 메모리(41)를 갖는다. 서브 프로세서 코어(5)는 제2 메모리(51)를 갖는다.
- [0122] 제1 메모리(41) 및 제2 메모리(51)에서, 제1 메모리(41) 및 제2 메모리(51)의 데이터의 콘텐츠들을 동기화하는 공유 메모리(411) 및 공유 메모리(511)가 각각 생성된다. 즉, 제1 메모리(41)에는, 공유 메모리(411)가 생성된다. 제2 메모리(51)에는, 공유 메모리(511)가 생성된다. 공유 메모리(411) 및 공유 메모리(511)는 미리 결정된 타이밍에 서로 데이터의 콘텐츠들을 동기화한다. 구체적으로는, 본 예시적인 실시예에서, 서브 프로세서 코어(5)가 실행을 정지하는 동안, 메인 프로세서 코어(4)가 공유 메모리(411)의 데이터를 및 공유 메모리(511)의 데이터와 동기화한다. 예를 들어, 이러한 처리를 통해, 공유 메모리(411)의 공유 메모리(511)와의 동기화가 수행된다.
- [0123] 전술한 바와 같이, 본 예시적인 실시예에서 정보 처리 장치는 운영 체제가 통합되어 있는 메인 프로세서 코어(4), 및 메인 프로세서 코어(4)에 의해 제어되고 운영 체제가 통합되어 있지 않은 서브 프로세서 코어(5)를 갖는다. 또한, 메인 프로세서 코어(4)에 의해 포함되는 제1 메모리(41)에는 공유 메모리(411)가 생성되고, 서브 프로세서 코어(5)에 의해 포함되는 제2 메모리(51)에는 공유 메모리(511)가 생성된다. 정보 처리 장치는 서브

프로세서 코어(5)가 실행을 정지하는 동안, 메인 프로세서 코어(4)가 공유 메모리(411)의 데이터를 공유 메모리(511)의 데이터와 동기화하도록 구성된다. 이러한 구성은 메인 프로세서 코어(4)에서 생성된 공유 메모리(411)를 서브 프로세서 코어(5)에서 생성된 공유 메모리(511)와 동기화할 수 있게 한다. 그 결과, 예를 들어 메인 프로세서 코어(4)의 프로세스와 서브 프로세서 코어(5)의 프로세스 사이에서 메모리 공유를 행하는 것이 가능하게 된다.

[0124] 전술한 정보 처리 장치가 동작할 때 실행되는 정보 처리 방법은, 제1 메모리(41)를 갖고 운영 체제가 통합되어 있는 메인 프로세서 코어(4); 제2 메모리(51)를 갖고 메인 프로세서 코어(4)에 의해 제어되고 운영 체제가 통합되어 있지 않은 서브 프로세서 코어(5)를 포함하는 정보 처리 장치에 의해 실행되는 정보 처리 방법이다. 제1 메모리(41) 및 제2 메모리(51)에는, 제1 메모리(41)와 제2 메모리(51) 사이의 데이터를 공유하는 공유 메모리(411) 및 공유 메모리(511)가 각각 생성된다. 정보 처리 방법은 메인 프로세서 코어(4)가 서브 프로세서 코어(5)가 실행을 정지하는 동안, 제1 메모리(41)에 생성된 공유 메모리(411)를 제2 메모리(51)에 생성된 공유 메모리(511)와 동기화하는 방법이다.

[0125] 전술한 구성을 갖는 정보 처리 방법의 발명은 또한 정보 처리 장치와 동일한 작용을 가져서 발명이 전술한 본 발명의 목적을 달성할 수 있다.

[0126] 또한, 예를 들어 운영 체제가 통합되어 있지 않은 서브 프로세서 코어(5)를 제어하기 위한 기능을 갖는 메인 프로세서 코어(4)도 본 발명의 목적을 달성할 수 있다(도 19 참조). 메인 프로세서 코어(4)는 제1 메모리(41)를 갖는다. 제1 메모리(41)에는 서브 프로세서 코어(5)에 의해 포함되는 제2 메모리(51)와 데이터를 공유하는 공유 메모리(411)가 생성된다. 메인 프로세서 코어(4)는 서브 프로세서 코어(5)가 실행을 정지하는 동안 제1 메모리(41)에 생성된 공유 메모리(411)를 제2 메모리(51)에 생성된 공유 메모리(511)와 동기화하는 공유 메모리 동기화 수단(42)을 갖는다.

[0127] 또한, 전술한 메인 프로세서 코어(4)는 메인 프로세서 코어(4)로 하여금 공유 메모리 동기화 수단(42)으로서 기능하게 하기 위한 프로그램을 설치하는 것에 의해 실현될 수 있다. 메인 프로세서 코어(4)는 제1 메모리(41)를 갖는다. 서브 프로세서 코어(5)에 의해 포함되는 제2 메모리(51)와 데이터를 공유하는 공유 메모리(411)가 제1 메모리(41)에 생성된다. 메인 프로세서 코어(4)는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어(5)를 제어하기 위한 기능을 갖는다. 공유 메모리 동기화 수단(42)은 서브 프로세서 코어(5)가 실행을 정지하는 동안 제1 메모리(41)에 생성된 공유 메모리(411)를 제2 메모리(51)에 생성된 공유 메모리(511)와 동기화한다.

[0128] 또한, 전술한 메인 프로세서 코어(4)가 동작할 때 실행되는 정보 처리 방법은 운영 체제가 통합되어 있지 않은 서브 프로세서 코어(5)를 제어하기 위한 기능을 갖는 메인 프로세서 코어(4)에 의해 실행되는 정보 처리 방법이다. 이 방법에서는, 메인 프로세서 코어(4)에서 생성된 제1 메모리(41)에 생성된 공유 메모리(411)의, 서브 프로세서 코어(5)에서 생성된 제2 메모리(51)에서 생성된 공유 메모리(511)와의 동기화가 서브 프로세서 코어(5)가 실행을 정지하는 동안 수행된다.

[0129] 전술한 구성들을 갖는 프로그램 및 정보 처리 방법의 발명들은 또한 전술한 메인 프로세서 코어와 동일한 작용들을 가져서 발명들이 전술한 본 발명의 목적을 달성할 수 있다.

[0130] <부기들>

[0131] 전술한 예시적인 실시예의 전부 또는 일부는, 이하의 부기들로서 설명될 수 있지만, 이에 한정되지 않는다.

[0132] (부기 1)

[0133] 제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어를 포함하는 정보 처리 장치로서- 메인 프로세서 코어에는 운영 체제가 통합되어 있고, 서브 프로세서 코어에는 운영 체제가 통합되어 있지 않음 -,

[0134] 제1 메모리에 공유 메모리 영역이 형성되고, 제2 메모리에 공유 메모리 영역이 형성되고, 제1 및 제2 메모리들의 공유 메모리 영역들 내의 데이터가 동기화되고;

[0135] 메인 프로세서 코어는, 서브 프로세서 코어가 동작을 정지하는 동안, 제1 및 제2 메모리들에 형성된 공유 메모리 영역들 내의 데이터를 동기화하도록 구성된다.

[0136] (부기 2)

[0137] 부기 1에 따른 정보 처리 장치로서,

- [0138] 서버 프로세서 코어는, 메인 프로세서 코어에 미리 결정된 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하도록 구성되고;
- [0139] 메인 프로세서 코어는 서버 프로세서 코어에 의해 요청된 미리 결정된 정보 처리를 실행한 후, 공유 메모리 영역들 중 하나에서 미리 결정된 정보 처리를 통해 갱신된 데이터를 공유 메모리 영역들 중 다른 하나로 복사하고, 그 후 서버 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하도록 구성된다.
- [0140] (부기 3)
- [0141] 부기 1 또는 2에 따른 정보 처리 장치로서,
- [0142] 서버 프로세서 코어는, 메인 프로세서 코어에 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하도록 구성되고- 정보 처리는 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 것임 -, 및
- [0143] 메인 프로세서 코어는 서버 프로세서 코어에 의해 요청된, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 정보 처리를 실행한 후, 갱신 데이터를 제1 메모리에 형성된 공유 메모리 영역으로 복사하고, 그 후 서버 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하도록 구성된다.
- [0144] (부기 4)
- [0145] 부기들 2 또는 3에 따른 정보 처리 장치로서, 정보 처리는 시스템 호출이다.
- [0146] (부기 5)
- [0147] 부기들 1 내지 4 중 임의의 것에 따른 정보 처리 장치로서, 메인 프로세서 코어는 서버 프로세서 코어가 동작을 정지할 때, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터가 갱신되어 있는지 여부를 확인하고, 제2 메모리에 형성된 공유 메모리 영역 내의 갱신 데이터를 제1 메모리에 형성된 공유 메모리 영역으로 복사하도록 구성된다.
- [0148] (부기 6)
- [0149] 부기들 1 내지 5 중 임의의 것에 따른 정보 처리 장치로서, 메인 프로세서 코어는, 서버 프로세서 코어가 동작을 정지할 때, 제1 메모리에 형성된 공유 메모리 영역 내의 갱신 데이터를 제2 메모리에 형성된 공유 메모리 영역으로 복사하도록 구성된다.
- [0150] (부기 7)
- [0151] 부기들 1 내지 6 중 임의의 것에 따른 정보 처리 장치로서,
- [0152] 서버 프로세서 코어는, 메인 프로세서 코어에 공유 메모리 영역을 형성하도록 요청하고, 또한 동작을 정지하도록 구성되고;
- [0153] 메인 프로세서 코어는, 서버 프로세서 코어에 의한 요청에 응답하여, 제1 및 제2 메모리들 내에 공유 메모리 영역들을 형성하고, 그 후 서버 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하도록 구성된다.
- [0154] (부기 8)
- [0155] 제1 메모리를 갖는 메인 프로세서 코어 및 제2 메모리를 갖고 메인 프로세서 코어에 의해 제어되는 서버 프로세서 코어를 포함하는 정보 처리 장치에 의해 실행되는 정보 처리 방법으로서- 메인 프로세서 코어에는 운영 체제가 통합되어 있고, 서버 프로세서 코어에는 운영 체제가 통합되어 있지 않고, 제1 및 제2 메모리들 각각에 공유 메모리 영역들이 형성되고, 제1 및 제2 메모리들의 공유 메모리 영역들 내의 데이터는 동기화되고, 정보 처리 방법은,
- [0156] 메인 프로세서 코어에 의해, 서버 프로세서 코어가 동작을 정지하는 동안, 제1 및 제2 메모리들에 형성된 공유 메모리 영역들 내의 데이터를 동기화하는 단계를 포함한다.
- [0157] (부기 9)
- [0158] 부기 8에 따른 정보 처리 방법으로서,
- [0159] 서버 프로세서 코어에 의해, 메인 프로세서 코어에 미리 결정된 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하는 단계; 및

- [0160] 메인 프로세서 코어에 의해, 서브 프로세서 코어에 의해 요청된 미리 결정된 정보 처리를 실행한 후, 공유 메모리 영역들 중 하나에서 미리 결정된 정보 처리를 통해 갱신된 데이터를 공유 메모리 영역들 중 다른 하나로 복사하고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하는 단계를 포함한다.
- [0161] (부기 9-1)
- [0162] 부기 9에 따른 정보 처리 방법으로서,
- [0163] 서브 프로세서 코어에 의해, 메인 프로세서 코어에 정보 처리를 실행하도록 요청하고, 또한 동작을 정지하는 단계- 정보 처리는 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 것임 -; 및
- [0164] 메인 프로세서 코어에 의해, 상기 서브 프로세서 코어에 의해 요청된, 상기 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 정보 처리를 실행한 후, 정보 처리를 통해 갱신된, 상기 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 제1 메모리에 형성된 공유 메모리 영역으로 복사하고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하는 단계를 포함한다.
- [0165] (부기 10)
- [0166] 운영 체제가 통합되어 있는 메인 프로세서 코어로서- 메인 프로세서 코어는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어를 제어하기 위한 기능을 가짐 -,
- [0167] 제1 메모리가 포함되고;
- [0168] 제1 메모리에 공유 메모리 영역이 형성되고, 공유 메모리 영역 내의 데이터는 서브 프로세서 코어에 의해 포함된 제2 메모리 내의 데이터와 동기화되고;
- [0169] 공유 메모리 동기화 수단이 포함되고, 공유 메모리 동기화 수단은 서브 프로세서 코어가 동작을 정지하는 동안 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하도록 구성된다.
- [0170] (부기 10-1)
- [0171] 부기 10에 따른 메인 프로세서 코어로서, 서브 프로세서 코어에 의한 요청에 응답하여, 미리 결정된 정보 처리가 실행된 후, 공유 메모리 동기화 수단을 사용하는 것에 의해, 공유 메모리 영역들 중 하나에서 미리 결정된 정보 처리를 통해 갱신된 데이터가 공유 메모리 영역들 중 다른 하나로 복사되고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어가 실행된다.
- [0172] (부기 10-2)
- [0173] 부기 10-1에 따른 메인 프로세서 코어로서, 서브 프로세서 코어에 의한 요청에 응답하여, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 정보 처리가 실행된 후, 공유 메모리 동기화 수단을 사용하는 것에 의해, 정보 처리를 통해 갱신된 제2 메모리에 형성된 공유 메모리 영역 내의 데이터가 제1 메모리에 형성된 공유 메모리 영역으로 복사되고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어가 실행된다.
- [0174] (부기 11)
- [0175] 메인 프로세서 코어로 하여금 공유 메모리 동기화 수단으로서 기능하게 하기 위한 명령어들을 포함하는 프로그램으로서,
- [0176] 메인 프로세서 코어는 제1 메모리를 갖고, 서브 프로세서 코어를 제어하기 위한 기능을 갖고, 공유 메모리 영역이 제1 메모리에 형성되고, 공유 메모리 내의 데이터는 서브 프로세서 코어에 의해 포함되는 제2 메모리의 데이터와 동기화되고, 운영 체제가 메인 프로세서 코어에 통합되어 있고, 운영 체제가 서브 프로세서 코어에 통합되어 있지 않고;
- [0177] 공유 메모리 동기화 수단은 서브 프로세서 코어가 동작을 정지하는 동안 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하도록 구성된다.
- [0178] (부기 11-1)
- [0179] 부기 11에 따른 프로그램으로서, 서브 프로세서 코어에 의한 요청에 응답하여 미리 결정된 정보 처리의 실행 후, 공유 메모리 동기화 수단으로 하여금 공유 메모리 영역들 중 하나에서 미리 결정된 정보 처리를 통해 갱신된 데이터를 공유 메모리 영역들 중 다른 하나로 복사하고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위

한 제어를 실행하기 위한 명령어들을 포함한다.

- [0180] (부기 11-2)
- [0181] 부기 11 또는 11-1에 따른 프로그램으로서, 서브 프로세서 코어에 의한 요청에 응답하여, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 정보 처리의 실행 후, 공유 메모리 동기화 수단으로 하여금 정보 처리를 통해 갱신된, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 제1 메모리에 형성된 공유 메모리 영역으로 복사하고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하기 위한 명령어들을 포함한다.
- [0182] (부기 12)
- [0183] 운영 체제가 통합되어 있는 메인 프로세서 코어에 의해 실행되는 정보 처리 방법으로서, 메인 프로세서 코어는 운영 체제가 통합되어 있지 않은 서브 프로세서 코어를 제어하기 위한 기능을 갖고, 정보 처리 방법은 서브 프로세서 코어가 동작을 정지하는 동안, 메인 프로세서 코어에 의해 포함된 제1 메모리에 형성된 공유 메모리 영역 내의 데이터를 서브 프로세서 코어에 의해 포함된 제2 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화하는 단계를 포함한다.
- [0184] (부기 12-1)
- [0185] 부기 12에 따른 정보 처리 방법으로서, 서브 프로세서 코어에 의한 요청에 응답하여 미리 결정된 정보 처리를 실행한 후, 공유 메모리 영역들 중 하나에서 미리 결정된 정보 처리를 통해 갱신된 데이터를 공유 메모리 영역들 중 다른 하나로 복사하고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하는 단계를 포함한다.
- [0186] (부기 12-2)
- [0187] 부기 12 또는 12-1에 따른 정보 처리 방법으로서, 서브 프로세서 코어에 의한 요청에 응답하여, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 갱신하는 정보 처리를 실행한 후, 정보 처리를 통해 갱신된, 제2 메모리에 형성된 공유 메모리 영역 내의 데이터를 제1 메모리에 형성된 공유 메모리 영역으로 복사하고, 그 후 서브 프로세서 코어의 동작을 재개시키기 위한 제어를 실행하는 단계를 포함한다.
- [0188] (부기 13)
- [0189] 운영 체제가 통합되어 있는 메인 프로세서 코어에 의해 제어되는 서브 프로세서 코어로서- 서브 프로세서 코어에 운영 체제가 통합되어 있지 않음 -,
- [0190] 제2 메모리가 포함되고;
- [0191] 제2 메모리에 공유 메모리 영역이 형성되고, 공유 메모리 영역 내의 데이터는 메인 프로세서 코어에 의해 포함된 제1 메모리 내의 데이터와 동기화되고;
- [0192] 제2 메모리에 형성된 공유 메모리 영역 내의 데이터는 서브 프로세서 코어가 동작을 정지하는 동안 제1 메모리에 형성된 공유 메모리 영역 내의 데이터와 동기화된다.
- [0193] 전술한 예시적인 실시예들 및 부기들에서 설명된 프로그램은 저장 장치에 저장되거나, 컴퓨터 판독 가능한 기록 매체에 기록된다. 예를 들어, 기록 매체는 플래터블 디스크, 광 디스크, 광자기 디스크 및 반도체 메모리와 같은 휴대용 매체이다.
- [0194] 예시적인 실시예들을 참조하여 본 발명 설명되었지만, 본 발명은 이들 예시적인 실시예들에 한정되는 것은 아니다. 본 기술분야의 통상의 기술자에 의해 이해될 수 있는 본 발명의 형태 및 세부 사항들의 다양한 변경들이 본 발명의 범위 내에서 이루어질 수 있다.
- [0195] 본 발명은, 2015년 2월 19일자로 출원된 일본 특허 출원 제2015-030566호에 기초하여 우선권의 이익을 주장하는 것이며, 그 개시내용은 본 명세서에 그 전체가 참고로서 포함된다.

부호의 설명

- [0196] 1 노드, 제어 노드
- 2 노드, 계산 노드

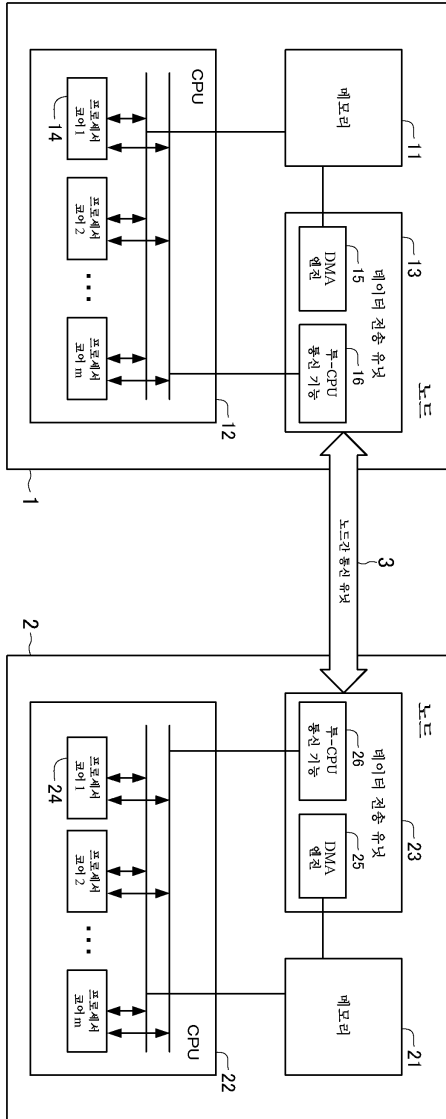
- 11, 21 메모리
- 12, 22 CPU
- 13, 23 데이터 전송 수단들
- 14, 24 프로세서 코어
- (24 계산 코어)
- 15,25 DMA 엔진
- 16, 26 투-CPU 통신 기능
- 240 범용 레지스터 세트
- 241 제어 레지스터 세트
- 242 예외 검출 수단
- 243 레지스터 세트 액세스 수단
- 244 예외 통지 수단
- 110 제어 노드 OS
- 1101 페이지 테이블
- 1102 페이지 테이블 엔트리
- 1103 메모리 속성 설정 수단
- 1104 시스템 호출
- 120 대체 프로세스
- 1201 메모리 갱신 대체 실행 수단
- 1202 메모리 매핑 대체 실행 수단
- 1203 공유 메모리 정보
- 111, 211 공유 메모리
- 130 계산 노드 관리 수단
- 1301 계산 노드 메모리 갱신 확인 수단
- 1302 공유 메모리 동기화 수단
- 1303 프로세스 대응 테이블
- 13031 대체 프로세스 ID
- 13032 계산 코어 번호
- 200 계산 프로세스
- 201 공유 메모리 갱신 식별자
- 210 프로세스 이미지
- 4 메인 프로세서 코어
- 41 제1 메모리
- 411 공유 메모리
- 42 공유 메모리 동기화 수단
- 5 서브 프로세서 코어

51 제2 메모리

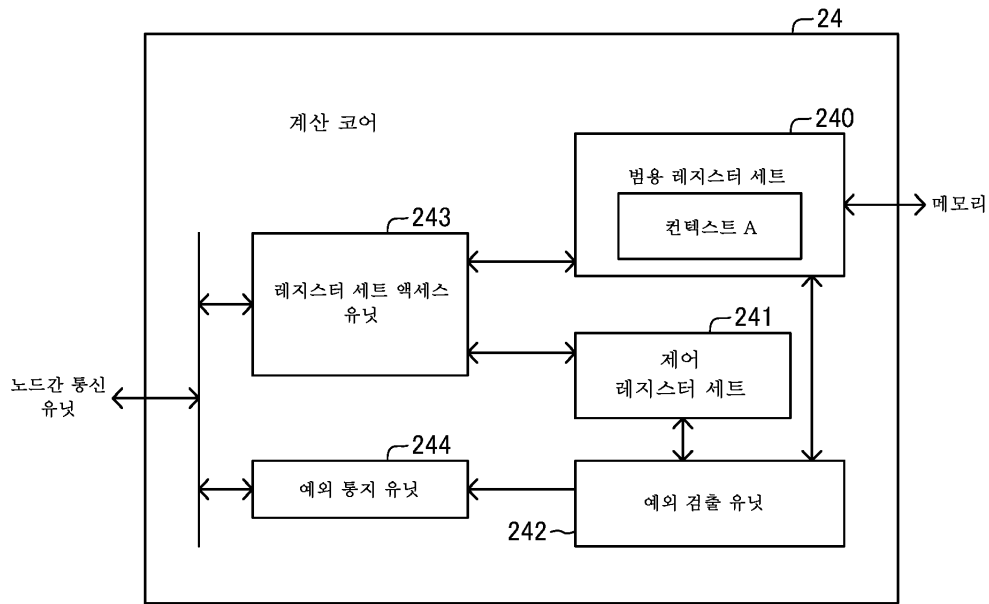
511 공유 메모리

도면

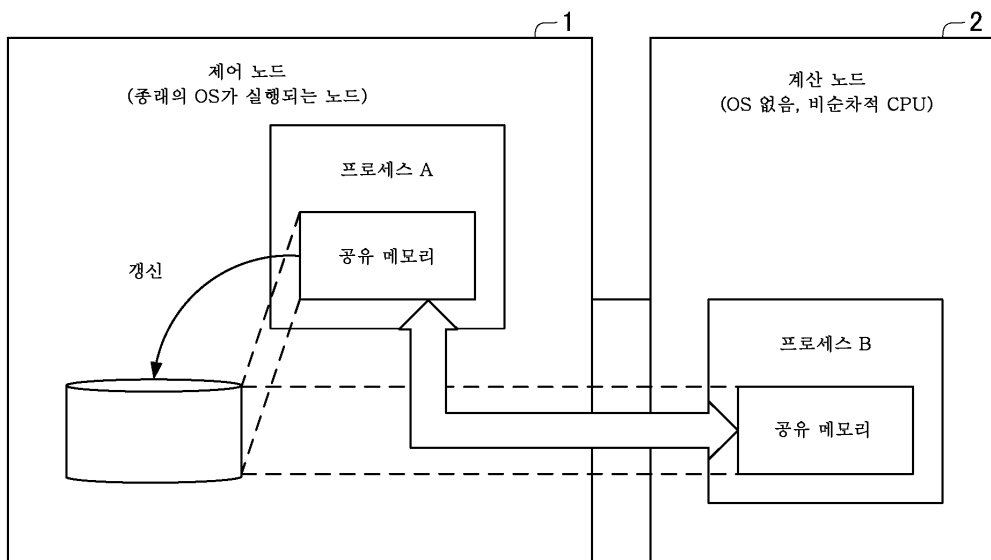
도면1



도면2

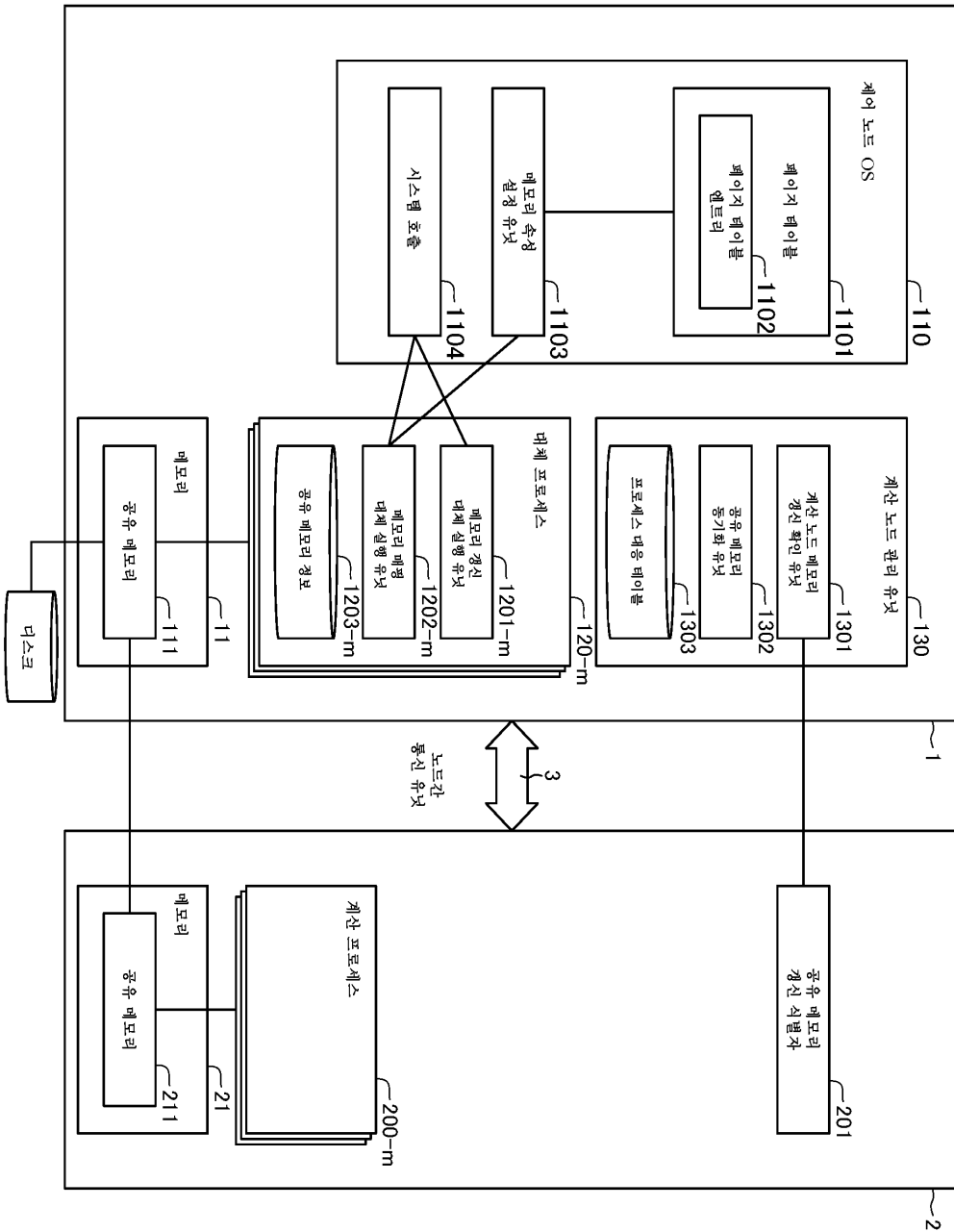


도면3

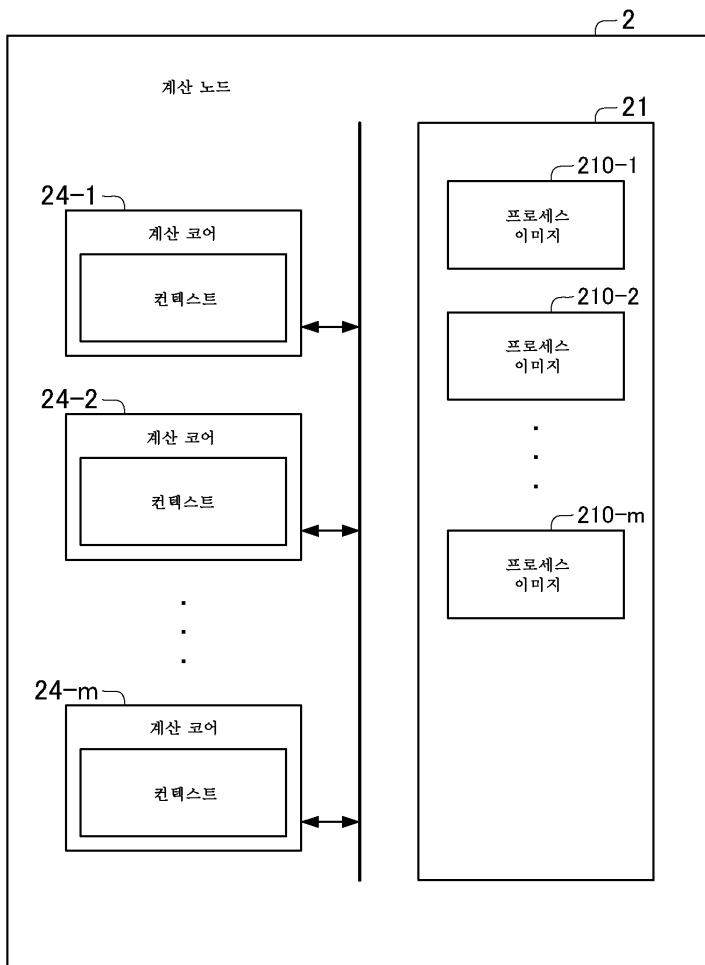


프로세스 A와 프로세스 B 사이에 메모리 콘텐츠를 동일하게 유지

도면4



도면5

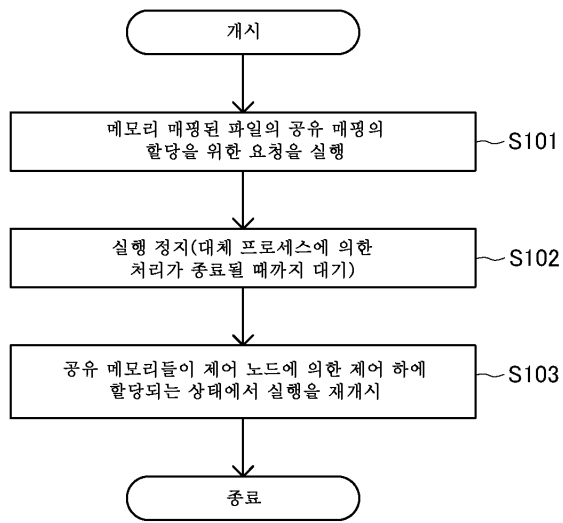


도면6

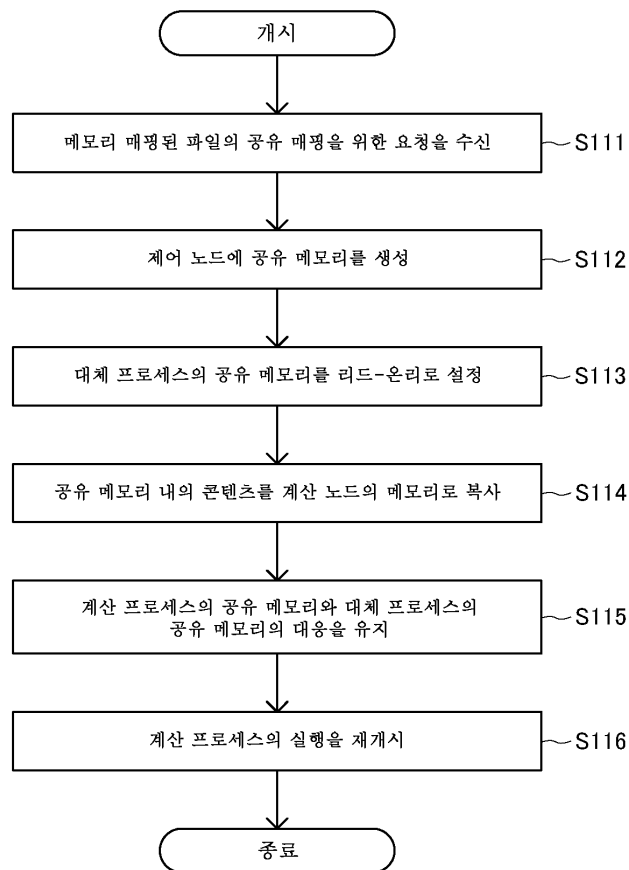
대체 프로세스 ID	계산 코어 번호
31996	코어 #0
32019	코어 #1
...	...

Arrows labeled 13031 and 13032 point to the '대체 프로세스 ID' and '계산 코어 번호' columns respectively.

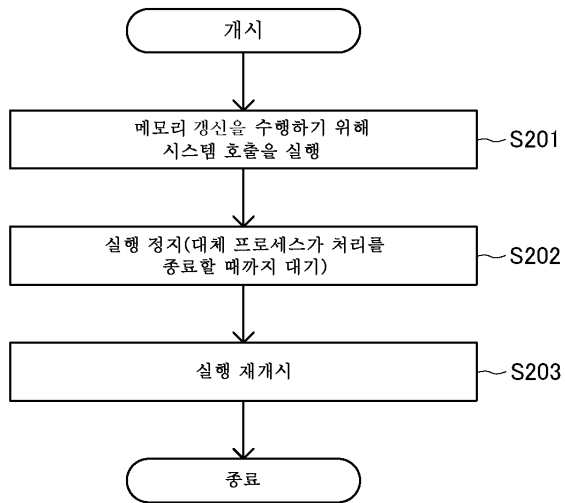
도면7



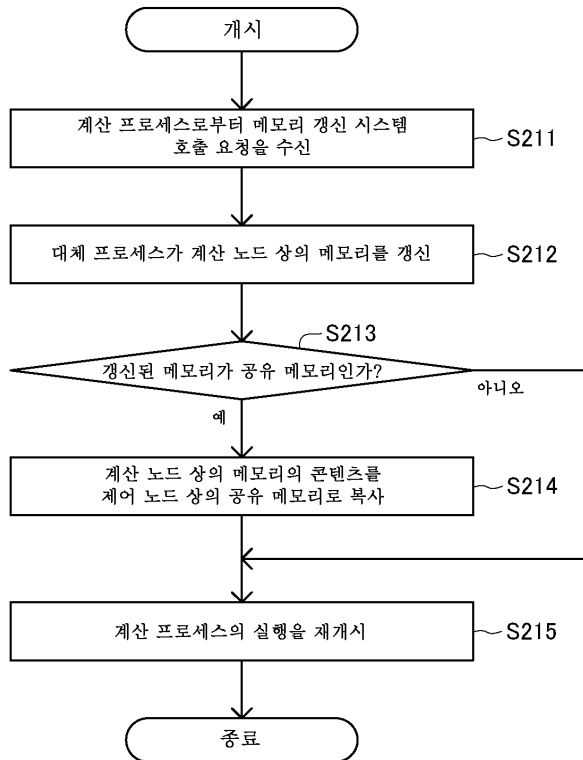
도면8



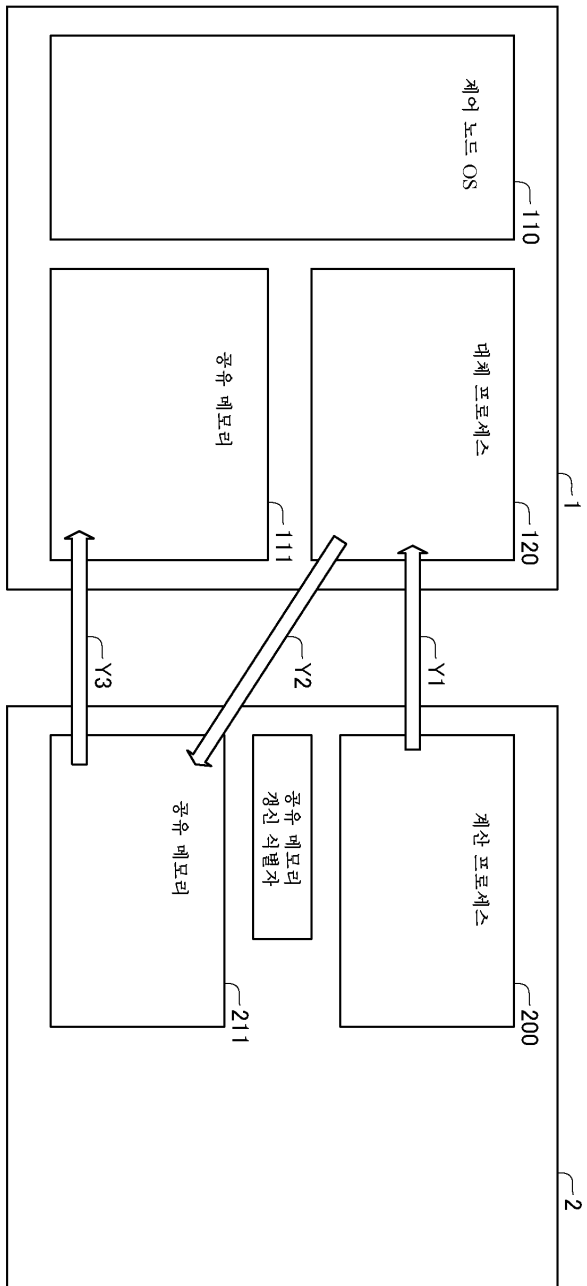
도면9



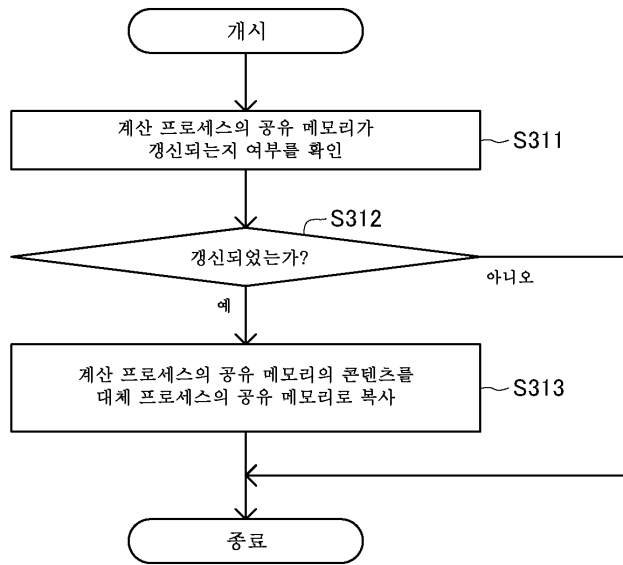
도면10



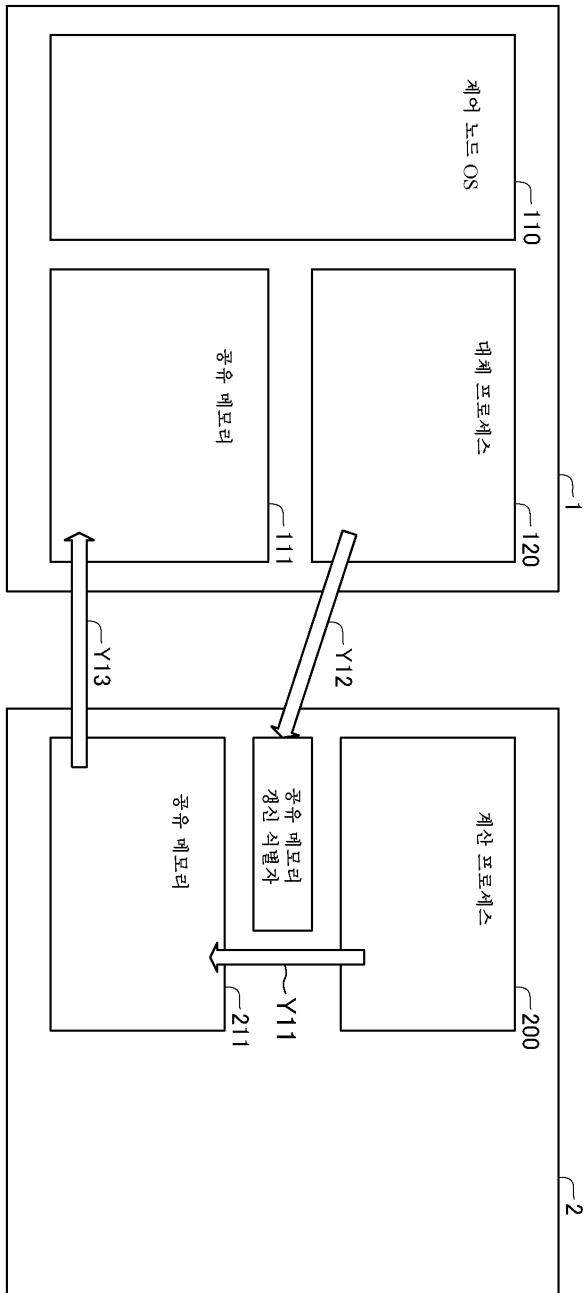
도면11



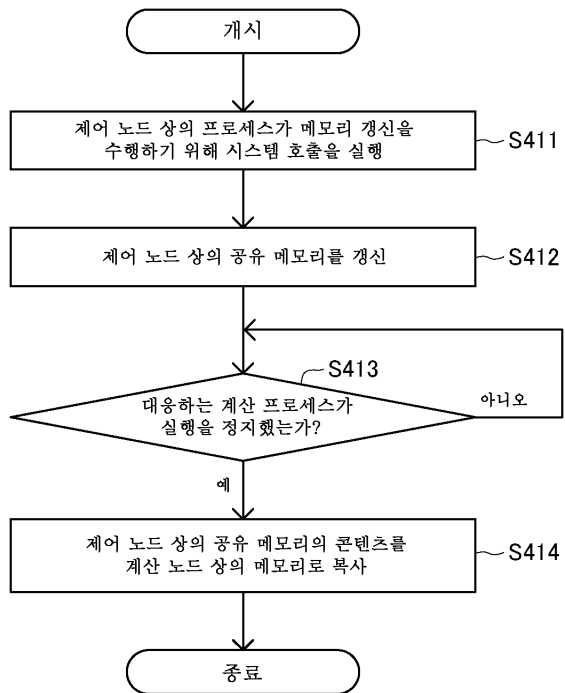
도면12



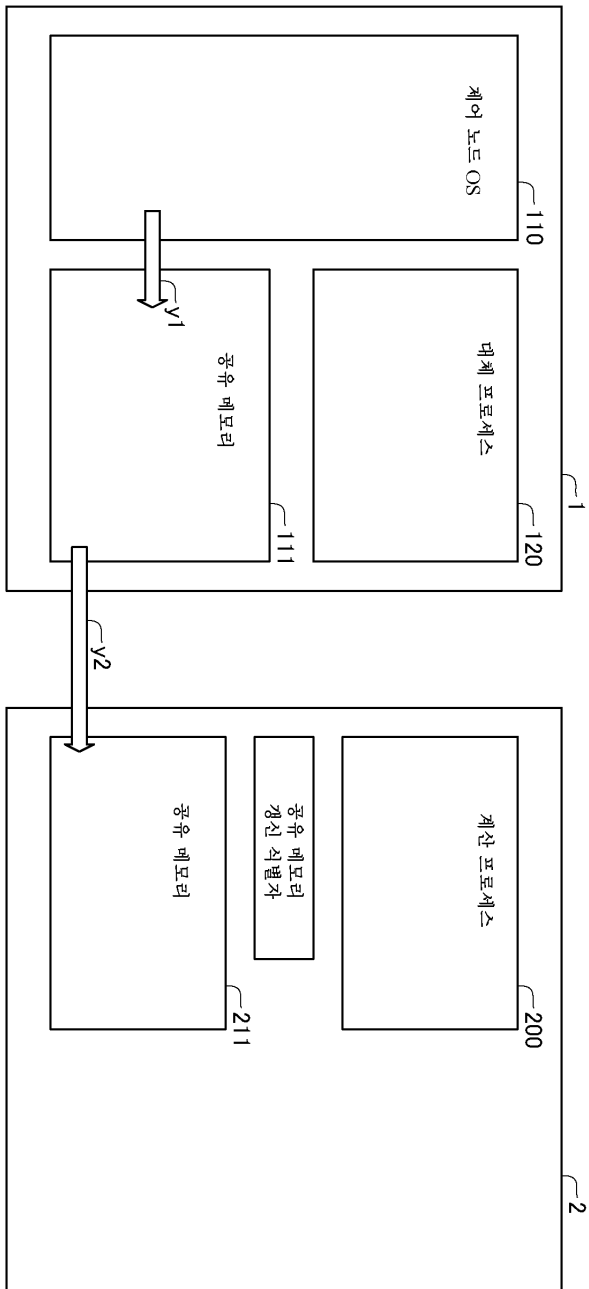
도면13



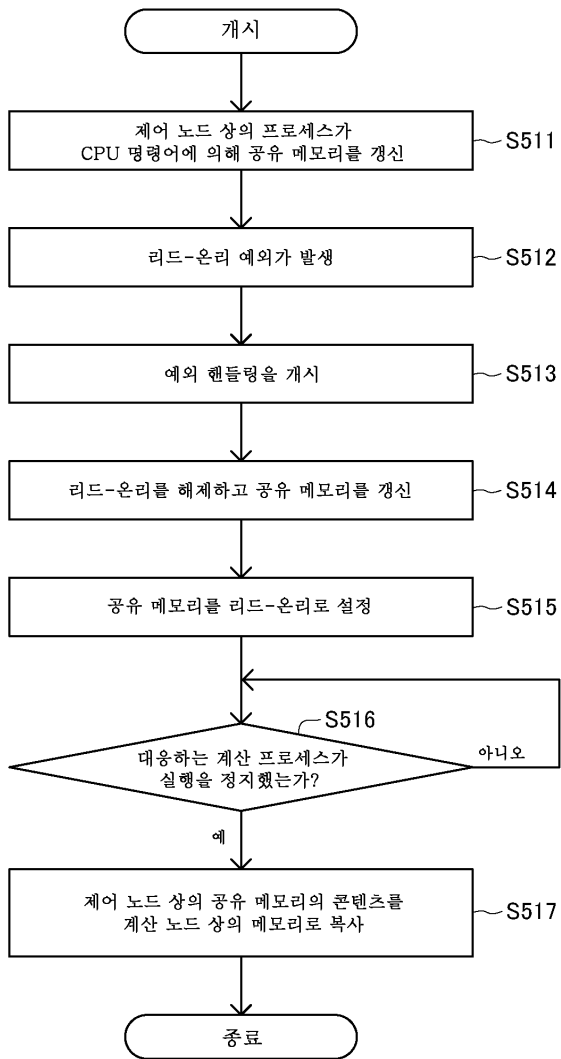
도면14



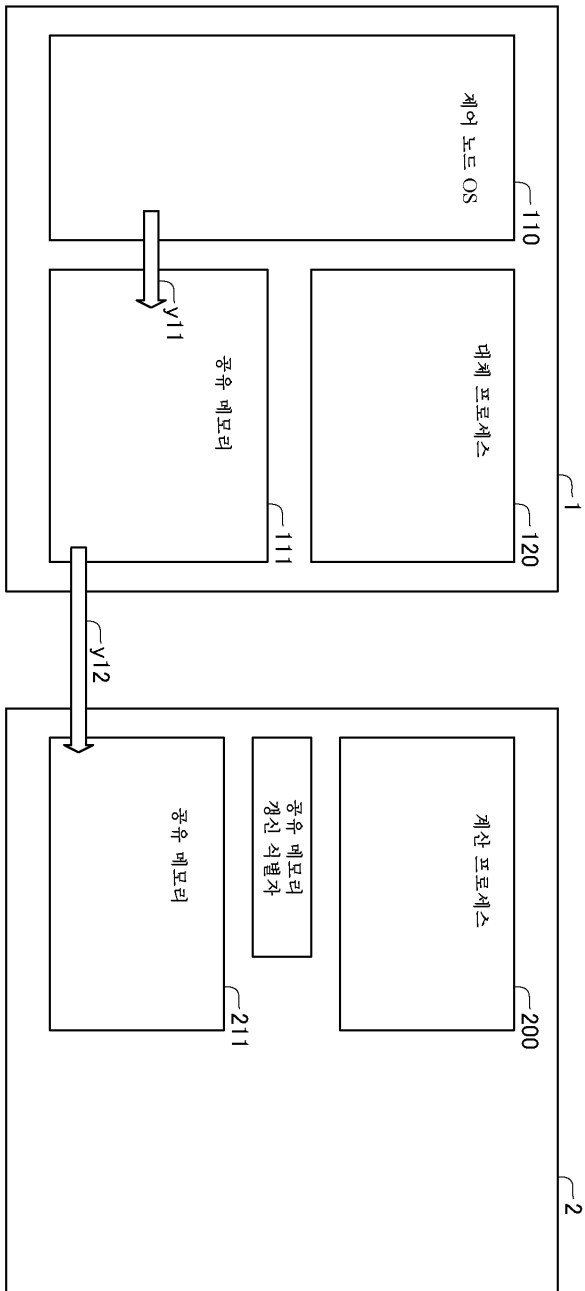
도면15



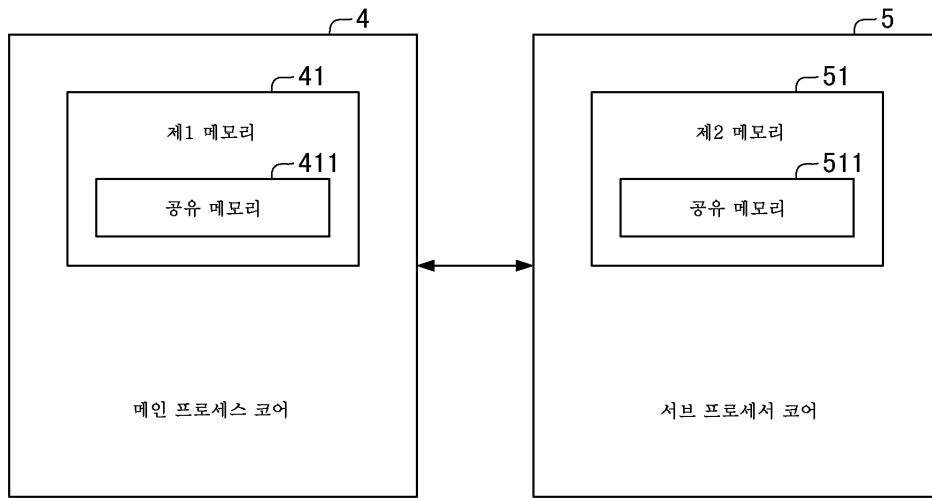
도면16



도면17



도면18



도면19

