(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl. [7]
G06F 15/00

(45) 공고일자 2002년11월18일
(11) 등록번호 10-0361635
(24) 등록일자 2002년11월06일

(21) 출원번호 10-2000-0024659
(22) 출원일자 2000년05월09일

(65) 공개번호 특2001-0007057
(43) 공개일자 2001년01월26일

(30) 우선권주장 09/314,214 1999년05월19일 미국(US)

(73) 특허권자
인터내셔널 비지네스 머신즈 코포레이션
미국 10504 뉴욕주 아몬크

(72) 발명자
암스트롱트로이데이비드
미국미네소타주55901로체스트8스트릿엔.더블유.3576
암스트롱윌리암조셉
미국미네소타주55944카손이스트올리브레인35
코리간마이클조셉
미국미네소타주55906로체스트20애비뉴엔.이.1510
나야르나레쉬
미국미네소타주55901로체스터벨모랄레인엔.더블유.5233
샌드토마스렘버트
미국미네소타주55901로체스터힐스보로레인엔.더블유.6004
쉬일제프리제이
미국미네소타주55901로체스터24애비뉴엔.더블유.6002
칼에릭
미국미네소타주55901-5202로체스터17스트릿엔.더블유.525
보센케네스차알즈
미국미네소타주55944카손박스66루럴루트3

(74) 대리인
김성택
조태연
김함곤
허정훈

심사관 : 이은철

(54) 논리적 분할부 관리기 및 방법

요약

컴퓨터 시스템에서 논리적 분할부(partitions)를 관리하는 분할부 관리기는 논리적 분할부들중 하나에서 저레벨 운용 시스템 코드에 훅(hook)을 포함한다. 컴퓨터 시스템의 자원을 관리하는 데 운용 시스템 코드를 사용함에 의해, 운용 시스템에서 이루어지는 어떤 변화라도 분할부 관리기의 기능에 자동 반영된다. 또한, 새로운 분할부 관리기를 생성할 때 양호하게 디버그되고 시험되는 운용 시스템의 저레벨 기능이 사용되어, 분할부 관리기를 생성할 때 드는 비용 및 시간을 감소시킬 수 있다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 본 발명의 양호한 실시예에 따른 논리적 분할(partitioning)을 지원하는 컴퓨터 장치의 블록도.

도 2는 2개의 논리적 분할부를 갖는 간단한 종래 기술의 시스템을 도시하는 블록도.

도 3은 2개의 논리적 분할부를 갖는 양호한 실시예에 따른 시스템을 도시하는 블록도.

도 4는 컴퓨터 시스템을 논리적으로 분할하는 공지된 방법을 예시하는 흐름도.

도 5는 본 발명의 양호한 실시예에 따른 컴퓨터 시스템을 논리적으로 분할하는 방법을 예시하는 흐름도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 특허 출원은 다음의 특허 출원들: 암스트롱 등에 의해 1999년 5월 19일에 출원된 출원번호 제 호인 "컴퓨터 시스템의 하나의 논리적 분할부에서의 최대 인터랙티브 작업을 나머지 분할부들에서의 최대 인터랙티브 작업과 별도로 지정하는 장치 및 방법"이라는 명칭의 미국 특허출원; 암스트롱 등에 의해 1999년 5월 19일에 출원된 출원번호 제 호인 "메모리 액세스 인터럽트를 통해서 발생된 처리기 리셋"이라는 명칭의 미국 특허출원; 암스트롱 등에 의해 1999년 5월 19일에 출원된 출원번호 제 호인 "논리적으로 분할된 컴퓨터에서 동시 사용 라이센스 관리"라는 명칭의 미국 특허출원; 암스트롱 등에 의해 1999년 5월 19일에 출원된 출원번호 제 호인 "논리적으로 분할된 컴퓨터용의 이벤트-구동형 통신 인터페이스"라는 명칭의 미국 특허출원과 관련되어 있다.

본 발명은 데이터 처리에 관한 것으로, 좀 더 구체적으로는 컴퓨터 시스템의 논리적 분할에 관한 것이다.

컴퓨터 시대가 도래함에 따라, 컴퓨터 시스템들은 다수의 다른 세팅에서 발견될 수 있는 매우 복잡한 장치들로 발전되어 왔다. 컴퓨터 시스템들은 하드웨어(예를 들어, 반도체, 회로판 등) 및 소프트웨어(예를 들어, 컴퓨터 프로그램)의 결합을 통상적으로 포함한다. 반도체 처리 및 컴퓨터 구조가 진보됨에 따라 컴퓨터 하드웨어의 성능이 향상되고, 더 복잡한 컴퓨터 소프트웨어가 향상된 하드웨어 성능을 이용하도록 발전되어와서, 결과적으로 현재의 컴퓨터 시스템이 수년

전보다 더 강력한 효율을 갖게 되었다.

특정한 컴퓨터 시스템의 계산 환경은 하드웨어 및 소프트웨어를 결합시킴으로써 형성된다. 그러므로, 다른 하드웨어 플랫폼 및 다른 운용 시스템은 다른 계산 환경을 제공한다. 최근에, 컴퓨터 시스템 자원을 다른 계산 환경으로 논리적으로 분할함에 의해 다른 계산 환경을 같은 물리적인 컴퓨터 시스템상에 구비하는 것이 가능하다는 것을 기술자들은 이해하게 되었다. IBM에 의해 개발된 AS/400 컴퓨터 시스템은 논리적 분할을 지원하는 컴퓨터 시스템의 한 예이다. AS/400상에서 논리적 분할이 이루어지길 원한다면, 분할부 관리기 코드[AS/400 용어로 "하이퍼비져 (hypervisor)"로 불리우는]는 다른 계산 환경을 같은 플랫폼상에서 형성하도록 설치된다. 분할부 관리기가 일단 설치되면, 논리적 분할부가 다른 계산 환경에서 생성될 수 있다. 분할부 관리기는 논리적 분할부에 의해 형성되는 개별적인 계산 환경을 유지하면서 컴퓨터 시스템에서 필요한 자원을 공유할 수 있도록 논리적 분할부를 관리한다.

논리적 분할부를 갖지 않은 컴퓨터 시스템에서는 분할부 관리기가 필요없고, 운용 시스템은 컴퓨터 시스템 자원에 대한 액세스를 제어한다. 하나 이상의 논리적 분할부를 갖는 컴퓨터 시스템에서, 분할부 관리기는 컴퓨터 시스템 자원에 대한 액세스를 제어하고, 각 논리적 분할부에 설치된 운용 시스템은 분할부 관리기를 사용하여 그 기능을 수행한다. 운용 시스템 및 분할부 관리기 모두가 컴퓨터 시스템에서 자원을 관리할 수 있어야하므로, 운용 시스템의 저레벨 기능이 분할부 관리기에서 또한 일반적으로 구현된다. 공지된 시스템에서, 분할부 관리기는 운용 시스템과 별개이고, 분할부 관리기는 운용 시스템과 일반적으로 별도로 유지된다. 그들이 비슷한 기능을 공유하게 되므로, 하나의 공통 기능이 변화하면, 그에 따라 다른 하나도 일반적으로 변화되어야 한다. 더 효율적인 방법으로 논리적 분할을 지원하는 메카니즘없이, 분할부 관리기의 프로그래밍과 운용 시스템 및 분할부 관리기의 개별적인 유지는 컴퓨터 산업에서 자원의 낭비를 계속해서 초래한다.

발명이 이루고자 하는 기술적 과제

본 발명에 따라, 컴퓨터 시스템에서 논리적 분할부를 관리하는 분할부 관리기는 논리적 분할부들중 하나에서 저레벨 운용 시스템 코드에 훅(hooks)를 포함한다. 컴퓨터 시스템의 자원을 관리하는 데 운용 시스템 코드를 사용함에 의해, 운용 시스템에서 이루어지는 어떤 변화라도 분할부 관리기의 기능에서 자동 반영된다. 또한, 새로운 분할부 관리기를 생성할 때 양호하게 디버그되고 시험되는 운용 시스템의 저레벨 기능이 사용되어, 분할부 관리기를 생성하는 데 드는 비용 및 시간을 감소시킬 수 있다.

본 발명의 앞선 특징 및 다른 특징 및 장점은 첨부 도면에서 예시했듯이 본 발명의 양호한 실시예에서 이하의 특정한 설명에 의해 명백해진다.

본 발명의 양호한 실시예는 같은 구성요소를 같은 도면번호로 표시하는 첨부 도면과 함께 이하에서 설명된다.

발명의 구성 및 작용

본 발명의 양호한 실시예에 따르면, 논리적 분할을 지원하는 컴퓨터 시스템은 논리적 분할부들중 하나의 운용 시스템에 후크를 사용하는 분할부 관리기를 포함하여 컴퓨터 시스템의 저레벨 제어를 수행한다. 운용 시스템 기능을 사용해서, 분할부 관리기의 크기가 매우 감소되고, 분할부 관리기의 유지 능력이 상당히 향상된다.

도1 에서, 컴퓨터 시스템(100)은 향상된 IBM AS/400 컴퓨터 시스템이고, 본 발명의 양호한 실시예에 따라 논리적 분할을 지원하는 하나의 알맞은 형태의 컴퓨터 시스템을 도시한다. 본 발명의 메카니즘 및 장치가 논리적 분할부를 지원

하는 어떤 컴퓨터 시스템에도 동일하게 적용된다는 것을 당업자는 알 수 있다. 도1 에 도시했듯이, 컴퓨터 시스템(100)은, 주 메모리(120)에 연결된 처리기(110)와, 대용량 저장 인터페이스(130)와, 터미널 인터페이스(140)와, 망 인터페이스(150)를 구비한다. 상기 시스템 구성요소들은 시스템 버스(160)를 사용해서 상호 연결된다. 대용량저장 인터페이스(130)는 대용량 저장 장치[직접 액세스 저장 장치(155) 등과 같은]를 컴퓨터 시스템(100)에 연결하도록 사용된다. 하나의 지정된 형태의 직접 액세스 저장 장치는 데이터를 CD-ROM(195)으로부터 판독할 수 있는 CD-ROM 드라이브이다.

주 메모리(120)는 하나의 1차 분할부(124) 및 하나 이상의 2차 분할부(126)를 관리하는 분할부 관리기(122)를 포함하고, 그 둘은 논리적 분할부들이다. 1차 분할부(124)는 운용 시스템(125)을 포함하고, 2차 분할부(126)가 운용 시스템(127)을 포함한다. 분할부 관리기(122)는 1차 분할부(124)에서 운용 시스템(125)에 대한 인터페이스(123)를 포함한다. 상기 방식으로 분할부 관리기(122)는 그 코드를 복제하는 대신에 운용 시스템(125)내에 이미 존재하는 코드를 사용하여 그 분할부 관리 기능을 수행한다.

운용 시스템(125)은 OS/400으로서 업계에서 공지된 멀티태스킹 (multitasking) 운용시스템이나, 본 발명의 정신 및 범위가 어떤 하나의 운용 시스템으로 제한되지 않음을 당업자는 인식하여야한다. 임의의 적당한 운용 시스템이 사용될 수도 있다. 운용 시스템(125)은 컴퓨터 시스템(100)의 자원을 관리하도록 저-레벨 코드를 포함하는 복잡한 프로그램들이다. 그 자원의 일부는, 처리기(110)와, 주 메모리(120)와, 대용량 저장 인터페이스(130)와, 터미널 인터페이스(140)와, 망 인터페이스(150)와, 시스템 버스(160)이다. 각 2차 분할부(126)의 운용 시스템(127)은 1차 분할부(124)의 운용 시스템(125)과 같게 될 수 있거나 완전히 다른 운용 시스템일 수 있다. 그러므로, 1차 분할부(124)가 OS/400 운용 시스템을 구동할 수 있는 반면에, 2차 분할부(126)는 가능하게는 다른 판(release)이거나 다른 환경 설정(예를 들어, 시간 대역)을 갖는 OS/400의 다른 예를 구동할 수 있다. 2차 분할부(126)의 운용 시스템(127)이 하드웨어 및 운용 시스템 인터페이스와 호환된다면 OS/400과도 다를 수 있다. 상기 방식으로 논리적 분할부는 동일한 물리적인 컴퓨터 시스템상에서 완전히 다른 계산 환경을 제공할 수 있다.

주 메모리(120)내의 분할부(124 및 126)가 도1 에 도시된다. 그러나, 분할부가 메모리를 제외한 자원을 포함하는 논리적 구조라는 것을 당업자는 인식하게 된다. 논리적 분할부가 하나 이상의 처리기 및 나머지 시스템 자원을 배정함과 함께 메모리중 일부를 통상적으로 지정한다. 그러므로, 1차 분할부(124)는 대용량저장 인터페이스(130), 터미널 인터페이스(140), 망 인터페이스(150), 또는 나머지 I/O 장치에 대한 인터페이스의 기능을 제공할 수 있는 하나 이상의 I/O 처리기와 함께 2개의 처리기들 및 메모리(120)중 일부를 포함하여 형성될 수 있다. 그후, 2차 분할부(126)는 3개의 다른 처리기, 메모리(120)중 다른 부분, 및 하나 이상의 I/O 처리기를 포함하여 형성될 수 있다. 도1 에 도시된 분할부는 컴퓨터 시스템(100)내의 메모리(120) 외부에 시스템 자원을 포함하는 논리적 분할부를 심벌로 나타낸다.

컴퓨터 시스템(100)은 컴퓨터 시스템(100)의 프로그램들로 하여금 주 메모리 (120) 및 DASD 장치(155) 등의 다수의 소형 저장 엔티티에 액세스하는 대신에 대형의 단일저장 엔티티에만 액세스하는 것처럼 동작하게 하는 공지된 가상 어드레싱 메카니즘을 사용한다. 그러므로, 분할부 관리기(122) 및 분할부(124 및 126)가 주 메모리(120)에 상주하도록 도시되어 있지만, 상기 아이템들이 동시에 주 메모리(120)에 모두 완전히 반드시 포함될 필요가 없다는 것을 당업자는 알 수 있다. 용어"메모리"가 컴퓨터 시스템(100)의 전체 가상 메모리로 본 명세서에서 일반적으로 불리워진다.처리기(110)는 하나 이상의 마이크로프로세서들 및/또는 집적 회로로 구성될 수 있다. 처리기(110)가 주 메모리(120)에 저장된 프로그램 명령을 실행한다. 주 메모리(120)는 처리기(110)가 액세스할 수 있는 프로그램 및 데이터를 저장한다. 컴퓨터 시스템(100)의 개시시, 처리기(110)가 분할부 관리기(122)를 구성하는 프로그램 명령을 초기에 실행시켜서, 논리적 분할부의 운용 시스템을 초기화한다.

컴퓨터 시스템(100)이 단일 처리기 및 단일 시스템 버스만을 포함하도록 도시되었지만, 본 발명이 다수의 처리기 및/또는 다수의 버스들을 갖는 컴퓨터 시스템을 사용해서 실시될 수 있음을 당업자는 알 수 있다. 또한, 본 발명의 양호한 실시예에서 사용되는 인터페이스들(AS/400 용어로 입력/출력 처리기로 불리우는) 각각은 계산-집중(compute-intensive) 처리를 처리기(110)로부터 오프로드시키기 위해 사용되는 개별적으로 완전히 프로그램된 마이크로프로세서들을 포함한다. 그러나, 본 발명이 I/O 어댑터를 비슷한 기능을 수행하기 위해 간단히 사용하는 컴퓨터 시스템에 동일하게 적용하는 것을 당업자는 알 수 있다.

터미널 인터페이스(140)는 하나 이상의 터미널(165)을 컴퓨터 시스템(100)에 직접 연결하기 위해 사용된다. 비-지능[즉, 더미(dumb)] 터미널 또는 완전히 프로그램가능한 워크스테이션일 수 있는 그 터미널(165)은 시스템 관리기 및 사용자로 하여금 컴퓨터 시스템(100)과 통신하도록 사용된다. 그러나, 터미널 인터페이스(140)가 하나 이상의 터미널(165)과의 통신을 지원하기 위해 구비되지만, 컴퓨터 시스템 (100)이 터미널(165)을 반드시 필요하지는 않는 데, 왜냐하면 사용자 및 다른 처리와의 모든 필요한 인터랙션은 망 인터페이스(150)를 경유해서 발생할 수 있기 때문이다.

망 인터페이스(150)가 나머지 컴퓨터 시스템 및/또는 워크스테이션(예를 들어, 도 1의 175)을 망(170)을 거쳐 컴퓨터 시스템(100)에 연결하기 위해 사용된다. 망 연결(170)이 현재의 아날로그 및/또는 디지털 기술을 사용하거나 미래의 망 메카니즘을 통해서 이루어지든간에 상관없이, 본 발명은 컴퓨터 시스템(100)이 나머지 컴퓨터 시스템 및/또는 워크스테이션에 어떻게 연결될 지라도 동일하게 적용된다. 또한, 다수의 다른 망 프로토콜은 망을 구현시키기 위해 사용된다. 상기 프로토콜은 컴퓨터로 하여금 망(170)을 거쳐 통신하게 하는 특수용 컴퓨터 프로그램이다. TCP/IP(Transmission Control Protocol/Internet Protocol)는 적당한 망 프로토콜의 예이다.

상기 관점에서, 본 발명이 충분히 기능적인 컴퓨터 시스템의 환경에서 설명되었고 계속 설명되겠지만, 본 발명이 다양한 형태로 프로그램 제품으로서 분배될 수 있고 그 분배를 실제로 수행하도록 사용된 신호 반송(bearing) 매체의 특정한 형태와 상관 없이 동일하게 적용됨을 당업자는 알 수 있다. 알맞은 신호 반송 매체의 예는 플로피 디스크 및 CD ROM[예를 들어, 도 1의 195] 등의 기록형 매체, 및 디지털 및 아날로그 통신 링크 등의 전송형 매체를 포함한다.

도 2에서, 컴퓨터 시스템(200)은 종래 기술을 사용해서 2개의 논리적 분할부로 분할부되는 것을 도시한다. 분할부 관리기(270)가 2개의 논리적 분할부(210 및 220)을 관리하도록 사용된다. 각 논리적 분할부는 대응하는 운용 시스템(230 및 240)을 포함하고, 각 운용 시스템은 운용 시스템의 저레벨 기능을 수행하는 커늘 (kernel)(250 및 260)을 포함한다. 운용 시스템 커늘(250)은 태스크 서포트(task support)(251), 동기 프리미티브즈(primitives)(252), 메인 저장 관리(253), 힙(heap) 관리(254), I/O 서브시스템(255), 및 사용자 인터페이스(256)을 포함한다. 비슷한 방식으로, 운용 시스템 커늘(260)은 대응하는 소자(261-266)를 또한 포함한다. 상기 아이템은 운용 시스템(230)을 참고로 이하에서 논의되는 데, 운용 시스템(230)의 커늘 기능에 대한 상기 설명이 운용 시스템(240)에도 적용되는 것을 알 수 있다.

태스크 서포트(251)는 태스크를 관리하는 운용 시스템(230)의 일부이다. 태스크는 수행되길 원하는 작업의 일부["작업"으로 공지된]이다. 태스트 서포트(251)는 필요할 때 시스템 자원을 할당하여 태스크를 수행하고, 그 태스크의 실행을 초기화한다. 동기 프리미티브즈(252)는 태스크들간에 통신하는 서포트를 제공하고, 메시지 큐, 공유된 데이터의 로크(locks), 및 태스크들간에 통신하는 나머지 메카니즘을 포함한다.

메인 저장 관리(253)는 분할부(210)에 대응하는 메인 메모리의 일부를 관리하는 운용 시스템(230)의 일부이다. 힙 관리(254)는 "힙"으로서 공지된 메모리에서 특정한 저장 영역을 관리한다. 힙은 동적으로 할당 및 재할당될 수 있는 저장이다. 운용 시스템(230)이 태스크용으로 할당된 메모리를 필요로 할 때, 힙 관리(254)가 힙의 일부를 태스크에 할당할 수 있고, 태스크가 수행된 후, 힙 관리(254)는 사용된 메모리를 힙으로 복귀시킨다.

I/O 서브시스템(255)은 컴퓨터 시스템의 입력/출력 장치를 제어하는 운용 시스템(230)의 일부이다. 사용자 인터페이스(256)는 사용자가 운용 시스템(230)과 인터랙트(interact)하는 인터페이스이다.

분할부 관리기(270)는 컴퓨터 시스템(200)의 저레벨 제어를 제공한다. 분할부관리기(270)가 태스크 서포트(271), 동기 프리미티브즈(primitives)(272), 메인 저장 관리(273), 힙(heap) 관리(274), I/O 서브시스템(275), 및 사용자 인터페이스(276)을 포함하고, 그들은 운용 시스템(230)의 커늘(250)에서 동일한 명칭의 소자(251-256)와 비슷하다. 분할부 관리기(270)가 그와 같은 기능을 구비할 필요성이 있는 이유는 명백한데, 즉 운용 시스템이 논리적 분할부에서 설치될 때, 그것은 더 이상 컴퓨터 시스템을 완전하게 제어하지 못하나, 나머지 논리적 분할부의 운용 시스템과의 제어를 공유해야하기 때문이다. 운용 시스템에 의해 이전에 수행되는 저레벨 기능은 분할부 관리기에 의해 현재 수행되야하고, 그 분할부 관리기가 논리적 분할부의 운용 시스템과 인터랙트하여 필요한 때에 요청된 기능을 수행한다. 또한, 분할부 관리기(270)는 운용 시스템에 의해 제공된 서비스를 알지 못하거나 그들을 불러내는 방법을 알지 못하므로, 그 서비스가 분할부 관리기 자체에 의해 제공되야한다. 그와 같은 이유로 인해, 분할부 관리기(270)는 운용 시스템(230)에서 이미 수행되었던 아이템(271-276)의 기능을 포함한다.

아이템(271-276)에 더해져서, 분할부 관리기(270)는 페이지 테이블 관리 (281), 분할부간 통신(282), 인터럽트 처리(283), I/O 프리미티브즈(284), 시스템 자원 관리(285), 및 자원 구성 관리(286)에 대한 메카니즘을 또한 포함한다. 페이지 테이블 관리(281)는 각 분할부가 자체 소유한 메인 메모리의 일부에만 액세스하게되는 메카니즘이다. 논리적 분할부가 메인 메모리의 페이지에 어드레스능력을 설정하기위해 페이지 테이블 관리 코드를 호출해야 한다. 분할부간 통신 메카니즘(282)이 컴퓨터 시스템(200) 외부로 이동해야하는 데이터없이 고속 통신을 논리적 분할부간에 제공한다. 인터럽트 처리(283)는 컴퓨터 시스템에서 인터럽트를 처리하는 메카니즘을 제공한다. I/O 프리미티브즈(284)는 입력/출력 장치를 구동하는 저레벨 코드를 제공한다. 시스템 자원 관리(285)는 논리적 분할부들간에 공유되길 원하는 시스템-광역(wide) 자원을 관리한다. 시스템 클럭은 시스템 자원(285)의 관리에 의해 관리되는 시스템 자원의 예이다. 자원 구성 관리(286)가 각 논리적 분할부에 대해 컴퓨터 시스템 자원의 구성을 결정한다. 환언해서, 자원 구성 관리(286)가 각 논리적 분할부에 갖고 있는 자원의 트랙을 유지한다.

도2 에 도시된 종래기술의 시스템(200)은 각 운용 시스템 커늘에서 상당한 코드량을 포함하고, 분할부 관리기(270)에서 다수의 비슷한 기능을 수행하는 코드를 제공한다. 분할부 관리기(270) 및 운용 시스템 커늘(250)에서 비슷한 기능을 제공하는 것은 새로운 하드웨어 버스 구조를 수용하도록 시스템 커늘(250)을 운용하는 I/O 서브시스템(255)이 변화하고 그 변화에 대응해서 분할부 관리기(270)의 I/O 서브시스템(275)에도 변화를 요구한다는 것을 의미한다. 그 비슷한 기능을 수행하는 2개의 다른 프로그램을 유지하는 것은 비슷한 기능에 대해 개발 및 시험 결과를 중복함에 의해 시간 및 자원의 낭비이다.

도3 에서, 본 발명의 양호한 실시예에 따른 컴퓨터 시스템(300)은 도3 에 도시된 논리적 분할부를 갖는 컴퓨터 시스템이고, 도1 의 도면번호(100)와 비슷하다. 1차 분할부(310)는 커늘(350)을 갖는 운용 시스템(330)을 구비한다. 커늘(350)은 도2 에 도시된 메카니즘(251-256)과 비슷한 메카니즘(351-356)을 포함한다. 커늘(350)이 OS 훅(357)을 통해서 운용 시스템을 호출함으로써 프로그램으로 하여금 운용 시스템에서 형성된 기능을 사용하게하는 운용 시스템 훅(357)을 구비한다. 분할부(310)가 "1차 분할부"로서 본 명세서에서 불리워지는 데 왜냐하면 그것은 소정의 기능을 수행하도록 분할부 관리기(370)에 의해 사용될 수 있는 운용 시스템(330)을 포함하기 때문이다. 모든 나머지 분할부가 "2차 분할부"로 본 명세서에서 불리운다.

분할부 관리기(370)는 도2 에 도시된 구성요소(281-286)와 비슷한 메카니즘 (381-386)을 포함한다. 또한, 분할부

관리기(370)는 분할부 관리기(370)로 하여금 논리적 분할부의 사용자수를 트랙킹하게하는 소프트웨어 라이센스 관리 프리미티브즈(387)를 포함하여 소프트웨어 라이센스를 실시한다. 분할부 관리기(370)는 OS 훅(357)을 호출하는 운용 시스템 인터페이스(368)를 포함하여 다수의 기능을 수행한다. 태스크 서포트, 동기 프리미티브즈, 메인 저장 관리, 힙 관리, I/O 서브시스템, 및 사용자 인터페이스를 구비하기위해 개별 코드를 분할부 관리기(370)내에 구비하는 대신에, 분할부 관리기(370)는 1차 분할부(310)에서 운용 시스템(330)의 커늘에 이미 상주하고 있는 메카니즘(351-356)을 사용한다. 이 방법은 몇개의 장점을 갖는다.

다수의 컴퓨터 시스템은 수년동안 정의가 명확하고 디버그되었던 운용 시스템을 갖는다. 신뢰할만한 커늘을 갖는 운용 시스템이 사용가능하다면, 프로그래머는 비슷한 기능들을 스크래치(scratch)로부터 재구현하느니 보다 정의가 명확하고 시험된 훅에 대한 호출을 운용 시스템내로 간단히 함에 의해 분할부 관리기(370)를 구현할 수 있다. 그러므로, OS 인터페이스(368) 및 OS 훅(357)을 구비함에 의해, 분할부 관리기(370)는 운용 시스템(330)에 이미 제공된 기능들로부터 장점을 얻을 수 있다. 그 장점은 분할부 관리기(370)를 구현하는 데 필요한 프로그래밍량을 대단히 감소시킨다는 것이고, 분할부 관리기(370)을 디버그하고 시험하는 데 필요한 시간을 대단히 감소시킨다는 것이다. 그러므로 본 발명은 컴퓨터 시스템용 분할부 관리기를 개발하는 비용을 적게 들게 한다.

운용 시스템(330)의 커늘(35)이 도3 에 없는 다수의 나머지 기능을 포함할 수 있고, 도시된 기능들의 서브세트를 포함한다. 도3 의 운용 시스템(330 및 340)의 기능들은 운용 시스템 커늘에 있는 다수의 가능한 기능들의 일부를 예시하는 예를 도시하고 있다. 또한, 분할부 관리기(370)의 기능(381-387)이 구현될 수 있는 알맞은 기능의 예로서 도시되고, 그 기능들의 서브세트뿐만 아니라 도시안된 다른 알맞은 기능들이 본 발명의 범위내에서 분할부 관리기(370)내에서 구현될 수 있다. 본 발명의 범위내에서 무수한 다른 기능들이 운용 시스템 커늘 또는 분할부 관리기내에서 결합될 수 있음을 당업자는 알 수 있다.

도3 의 컴퓨터 시스템(300)에서 분할부 관리기(370)가 1차 분할부(310)의 운용 시스템에 의해 제공된 기능들을 사용하고, 분할부 관리기(370) 및 논리적 분할부 (310 및 320)를 설치하는 방법은 종래 기술에서 공지된 방법과 다르다. 첫째로, 종래 기술의 방법(400)이 도4 에서 설명된다.

종래 기술에서, 하나 이상의 논리적 분할부가 컴퓨터 시스템상에서 생성되길 바란다면, 처음에 분할부 관리기가 설치된다(단계(410)). 다음에, 논리적 분할부가 생성된다(단계(420)). 그후 운용 시스템은 분할부상에서 설치된다(단계(430)). 운용 시스템이 모든 논리적 분할부상에 설치된다면(단계(440)=예), 그 설치 방법이 종료된다. 운용 시스템을 갖지 않는 논리적 분할부가 여전히 남아있다면(단계(440)=아니오), 운용 시스템은 다음의 논리적 분할부상에 설치되고(단계(430)), 운용 시스템이 모든 논리적 분할부에 설치될 때까지 단계(430 및 440)가 반복된다(단계(440)=예).

도5 에서, 본 발명에 따라 컴퓨터 시스템을 분할하는 방법(500)이 1차 분할부를 생성하고 분할부 관리기를 설치함에 의해 개시된다(단계(510)). 그후 분할부 관리기는 하나 이상의 2차 분할부를 생성하도록 사용된다(단계(520)). 그후, 운용 시스템은 2차 분할부에 설치된다(단계(530)). 운용 시스템이 모든 2차 분할부에 설치되었다면(단계(540)=540), 방법(500)이 종료된다. 운용 시스템을 갖지 않는 2차 분할부가 남아있다면, 모든 2차 분할부가 설치된 운용 시스템을 가질 때까지 단계 (530 및 540)가 반복된다(단계(540)=예).

도5 의 단계(510)에서 1차 분할부를 생성하고 분할부 관리기를 설치하는 이유는 본 발명의 양호한 실시예가 운용 시스템(330)의 일부로서 분할부 관리기(370)를포함하기 때문이다. 그러므로, 운용 시스템(330)을 설치하는 프로세스가

분할부 관리기(370)에 관한 정보용의 프롬프트를 또한 포함한다. 그것은 소프트웨어 설치기 (installer)가 운용 시스템 및 분할부 관리기(370) 모두를 설치하기위해 운용 시스템용으로 동일하며 친근한 인터페이스를 사용할 수 있다는 것을 의미한다. 같은 운용 시스템이 2차 분할부(320)상에 설치될 때 분할부 관리기 코드를 포함하나, 그 코드는 2차 분할부에 상주하고 있기 때문에 사용될 수 없음을 주목하자.

발명의 효과

상기 설명된 본 발명은 논리적 분할부에 상주하는 운용 시스템에 의해 제공된 기능을 사용하는 분할부 관리기를 구비한다. 상기와 같은 저레벨 기능을 수행하도록 운용 시스템에서 기존 코드를 사용함에 의해, 코드 복제는 방지되고, 코드 유지는 매우 개선된다. 또한, 분할부 관리기를 유지하는 사용자 인터페이스 및 절차는 1차 분할부의 운용 시스템을 유지하도록 사용한 것과 동일하여, 공지된 분할부 관리기를 유지하는 것과 비교해 본 발명의 분할부 관리기를 유지하는 것이 매우 간단함을 알 수 있다.

본 발명의 범위내에서 다양한 변형이 가능하다는 것을 당업자는 알 수 있다. 그러므로, 본 발명이 그 양호한 실시예를 참고로 도시 및 설명되었지만, 형태 및 상세한 내용의 변화가 본 발명의 정신 및 범위를 벗어남이 없이 이루어짐을 당업자는 이해하게된다.

(57) 청구의 범위

청구항 1.

논리적 분할부 관리기에 있어서,

적어도 하나의 처리기와,

상기 적어도 하나의 처리기에 결합된 메모리와,

운용 시스템 및 상기 적어도 하나의 처리기중 일부 및 상기 메모리의 일부를 각기 포함한, 상기 논리적 분할부 관리기에 형성된, 복수의 논리적 분할부와,

상기 복수의 논리적 분할부에 의해 컴퓨터 시스템 자원의 사용을 관리하고, 논리적 분할부중 기선택된 하나에 상주하고 있는 운용 시스템을 적어도 한번 호출하게 되는 분할부 관리기를 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 2.

제1항에 있어서, 상기 기선택된 논리적 분할부의 운용 시스템이 분할부 관리기를 적어도 한번 호출함에 의해 호출될 수 있는 훅을 포함하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 3.

제1항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 태스크 서포트 기능 및 동기 프리미티브즈(primitives)를 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 4.

제3항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 메인 저장 관리 기능 및 힙 관리 기능을 더 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 5.

제4항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 입력/출력(I/O) 서브시스템을 더 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 6.

제5항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 사용자 인터페이스를 더 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 7.

제6항에 있어서, 상기 분할부 관리기는 복수의 분할부에 걸쳐 사용자의 총수를 트랙킹하는 소프트웨어 라이센스 관리 프리미티브즈를 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 8.

제1항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 메인 저장 관리 기능 및 힙 관리 기능을 더 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 9.

제1항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 입력/출력(I/O) 서브시스템을 더 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 10.

제1항에 있어서, 상기 기선택된 논리적 분할부에 상주하는 운용 시스템이 분할부 관리기에 의해 호출될 수 있는 사용자 인터페이스를 더 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 11.

제1항에 있어서, 상기 분할부 관리기는 복수의 분할부에 걸쳐 사용자의 총수를 트랙킹하는 소프트웨어 라이센스 관리 프리미티브즈를 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 12.

논리적 분할부 관리기에 있어서,

적어도 하나의 처리기와,

상기 적어도 하나의 처리기에 결합된 메모리와,

운용 시스템 및 상기 적어도 하나의 처리기중 일부 및 상기 메모리의 일부를 각기 포함한, 상기 논리적 분할부 관리기에 형성된, 복수의 논리적 분할부와,

상기 복수의 논리적 분할부에 의해 컴퓨터 시스템 자원의 사용을 관리하고, 논리적 분할부들중 기선택된 하나에 상주하고 있는 운용 시스템을 적어도 한번 호출하게 되는 수단을 구비하는 것을 특징으로 하는 논리적 분할부 관리기.

청구항 13.

컴퓨터 시스템상에서 적어도 하나의 논리적 분할부를 생성하는 컴퓨터로 구현된 방법에 있어서,

(A) 1차 논리적 분할부를 생성하는 단계와,

(B) 상기 적어도 하나의 논리적 분할부에 의해 컴퓨터 시스템 자원의 사용을관리하고, 상기 1차 논리적 분할부에 설치되는 운용 시스템을 적어도 한번 호출하게 되는 분할부 관리기를 설치하는 단계와,

(C) 상기 운용 시스템을 1차 논리적 분할부에 설치하는 단계를 구비하는 것을 특징으로 하는 컴퓨터 시스템상에서 적어도 하나의 논리적 분할부를 생성하는 컴퓨터로 구현된 방법.

청구항 14.

제13항에 있어서,

(D) 나머지의 논리적 분할부를 생성하는 단계와,

(E) 단계(D)에서 생성된 상기 논리적 분할부에 운용 시스템을 설치하는 단계와,

(F) 컴퓨터 시스템에서 바라는 논리적 분할부의 갯수를 생성할 때까지 단계(D) 및 (E)를 반복하는 단계를 더 구비하는 것을 특징으로 하는 방법.

청구항 15.

복수의 논리적 분할부를 갖는 컴퓨터 시스템상에서 자원을 관리하는 컴퓨터로 구현된 방법에 있어서,

기형성된 논리적 분할부의 운용 시스템을 적어도 한번 호출하게되는 분할부 관리기를 구비하는 단계와,

상기 자원을 복수의 논리적 분할부에서 관리하고, 상기 운용 시스템을 적어도 한번 호출하게되는 분할부 관리기를 실행시키는 단계를 구비하는 것을 특징으로 하는 복수의 논리적 분할부를 갖는 컴퓨터 시스템상에서 자원을 관리하는 컴퓨터로 구현된 방법.

청구항 16.

컴퓨터 프로그램을 저장한 컴퓨터 판독 가능한 기록 매체에 있어서, 상기 컴퓨터 프로그램은,

복수의 논리적 분할부에 의해 컴퓨터 시스템 자원의 사용을 관리하고, 상기 논리적 분할부중 기선택된 하나에 상주하는 운용 시스템을 적어도 한번 호출하게되는 분할부 관리 수단을 구현하기 위한 명령어를 포함하는 것

인 컴퓨터 판독 가능 기록 매체.

청구항 17.

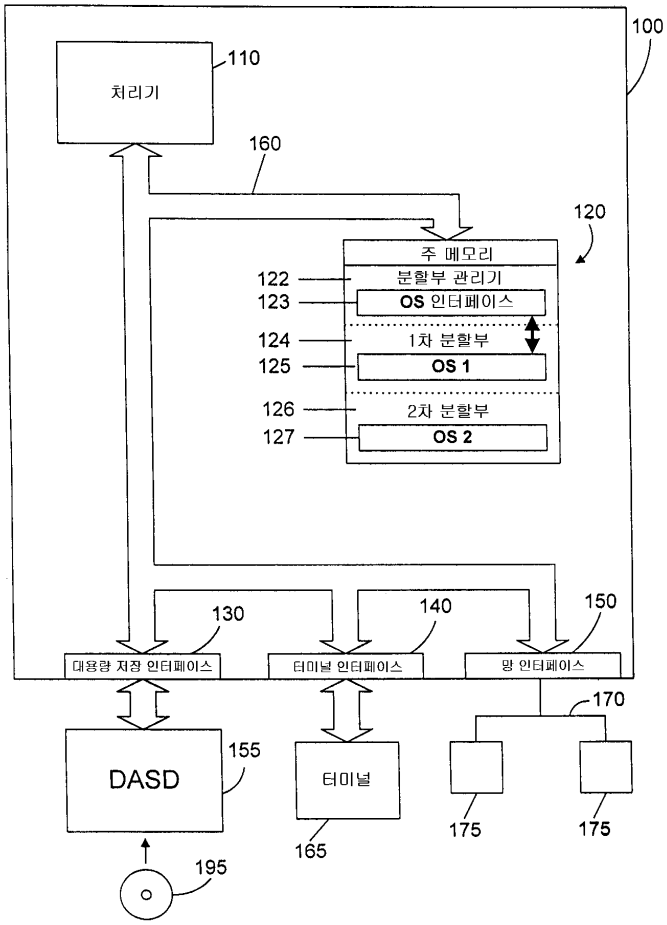
삭제

청구항 18.

삭제

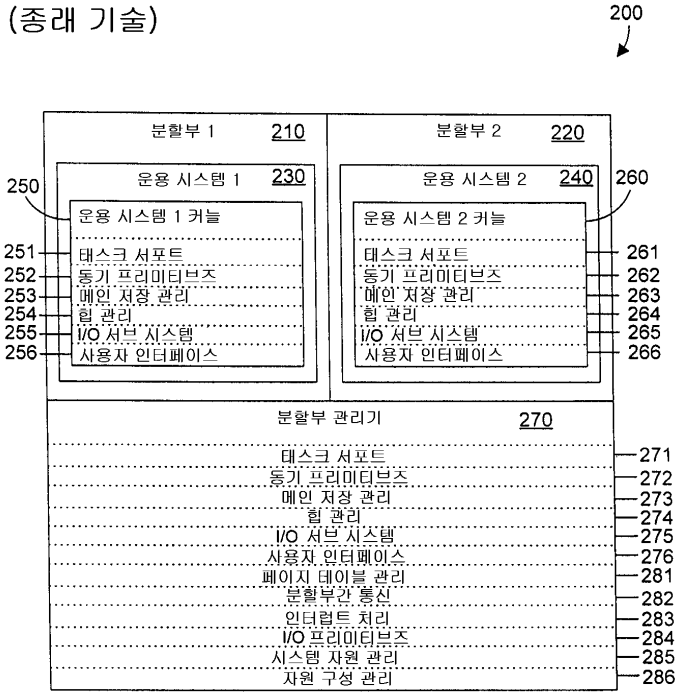
도면

도면 1

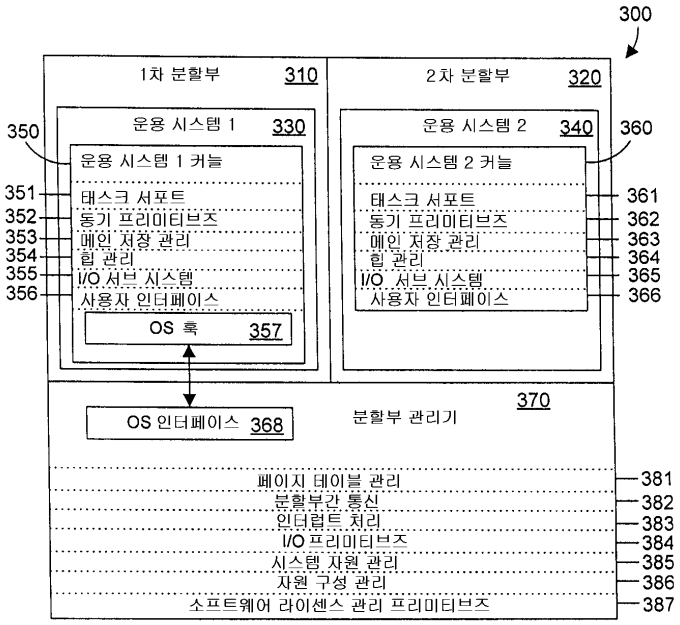

100
110
처리기
160
120
주 메모리
122
분할부 관리기
123
OS 인터페이스
124
1차 분할부
125
OS 1
126
2차 분할부
127
OS 2
130
140
150
대용량 저장 인터페이스
터미널 인터페이스
망 인터페이스
170
DASD
155
터미널
165
175
175
195

도면 2

(종래 기술)

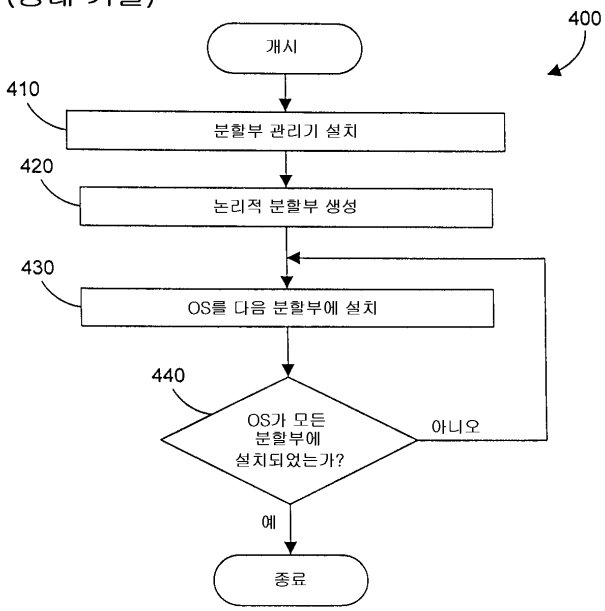

200
분할부 1 210
분할부 2 220
운용 시스템 1 230
운용 시스템 2 240
250
260
운용 시스템 1 커널
태스크 서포트
동기 프리미티브스
메인 저장 관리
힙 관리
I/O 서브 시스템
사용자 인터페이스
251
252
253
254
255
256
261
262
263
264
265
266
분할부 관리기 270
271
272
273
274
275
276
281
282
283
284
285
286
페이지 테이블 관리
분할부간 통신
인터럽트 처리
I/O 프리미티브스
시스템 자원 관리
자원 구성 관리

도면 3

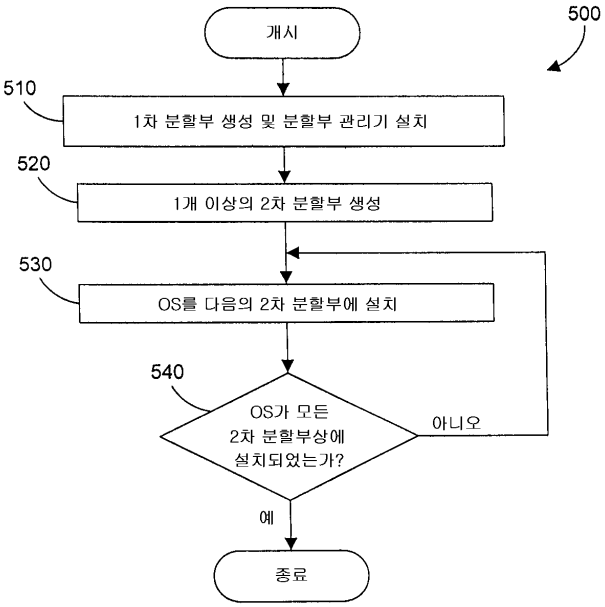

300
1차 분할부 310
2차 분할부 320
운용 시스템 1 330
운용 시스템 2 340
350
360
운용 시스템 1 커널
태스크 서포트
동기 프리미티브스
메인 저장 관리
힙 관리
I/O 서브 시스템
사용자 인터페이스
OS 훅 357
351
352
353
354
355
356
361
362
363
364
365
366
OS 인터페이스 368
370
분할부 관리기
381
382
383
384
385
386
387
페이지 테이블 관리
분할부간 통신
인터럽트 처리
I/O 프리미티브스
시스템 자원 관리
자원 구성 관리
소프트웨어 라이센스 관리 프리미티브스

도면 4

(종래 기술)


400
개시
410
분할부 관리기 설치
420
논리적 분할부 생성
430
OS를 다음 분할부에 설치
440
OS가 모든
분할부에
설치되었는가?
아니오
예
종료

도면 5


500
개시
510
1차 분할부 생성 및 분할부 관리기 설치
520
1개 이상의 2차 분할부 생성
530
OS를 다음의 2차 분할부에 설치
540
OS가 모든
2차 분할부상에
설치되었는가?
아니오
예
종료