(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl. [7]
H04M 3/00

(11) 공개번호 특2002-0036354
(43) 공개일자 2002년05월16일

(21) 출원번호 10-2000-0066497
(22) 출원일자 2000년11월09일

(71) 출원인 엘지전자주식회사
구자홍
서울시영등포구여의도동20번지

(72) 발명자 김태용
서울특별시양천구목2동534-13이정빌라302호

(74) 대리인 김영철

심사청구 : 없음

(54) 교환기에서 분산 시스템 시작 로더를 이용한 로딩 처리 방법

요약

본 발명은 교환기에서 분산 시스템 시작 로더를 이용한 로딩 처리 방법에 관한 것으로서, 교환기 내 장비의 부터(booter)가 시스템 관리 프로세서에 내장된 시스템 시작 로더로 로딩 요구 메시지를 송신하는 단계(a); 상기 단계(a)의 로딩 요구 메시지를 수신한 시스템 시작 로더가 메시지를 송신한 장비가 디바이스 프로세서나 데이터 제어 장치인지 여부를 판단하는 단계(b); 상기 단계(b)에서 메시지를 송신한 장비가 디바이스 프로세서나 데이터 제어 장치일 경우, 상기 디바이스 프로세서나 데이터 제어 장치가 속하는 SSP가 현재 운용중이거나 시동중인지 여부를 판단하는 단계(c); 상기 단계(c)에서 SSP가 운용중이거나 시동중일 경우, SSP 내의 분산 시스템 시작 로더로 로딩 요구 메시지를 전달하고, 상기 분산 시스템 시작 로더는 상기 단계(a)의 로딩 요구에 대한 응답 메시지를 부터로 전송하는 단계(d); 상기 분산 시스템 시작 로더가 상기 부터로부터 프로그램 블록 헤더 정보를 요구하는 메시지를 수신하면, 상기 요구한 프로그램 블록 헤더 정보를 상기 부터로 송신하는 단계(e); 및 상기 분산 시스템 시작 로더가 상기 부터로부터 프로그램 블록을 요구하는 메시지를 수신하면, 상기 요구한 프로그램 블록을 상기 부터로 송신하고, 하나의 블록에 대한 로딩 종료 메시지를 송신하는 단계(f)를 포함하는 것을 특징으로 한다.

본 발명에 의한 방법에 의하면, 분산 시스템 시작 로더에 의해 디바이스 프로세서와 데이터 제어 장치의 로딩을 분담하여 처리함으로써, 시스템 관리 프로세서의 과부하를 방지하고 IPC의 유실율 막을 수 있어 전체 교환기 로딩 속도의 향상 및 안정적인 응용 프로그램 블록의 실행을 보장할 수 있는 장점이 있다.

대표도

도 5

색인어
분산, 시스템 시작 로더, 교환기

명세서

도면의 간단한 설명

도 1은 종래 교환기에서의 로딩을 위한 시스템 구성을 도시한 것,

도 2는 본 발명의 바람직한 실시예에 따른 분산 시스템 시작 로더를 이용한 로딩 시스템의 구성을 도시한 것,

도 3은 종래 교환 시스템에서의 로딩 절차를 시스템 시작 로더와 부터별로 도시한 흐름도,

도 4는 본 발명의 바람직한 실시예에 따른 시스템 시작 로더와 부터의 동작에 대한 흐름도를 도시한 것,

도 5는 본 발명의 바람직한 실시예에 따른 분산 시스템 시작 로더 및 시스템 시작로더의 동작을 나타낸 흐름도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 교환기에서 분산 시스템 시작 로더를 이용한 로딩 처리 방법에 관한 것으로서, 더욱 상세하게는 스위칭 서브 시스템(Switching Sub System, 이하 " SSP" 라 함) 내에 분산 시스템 시작 로더를 실장하여 로딩 시 시스템 관리 프로세서가 모든 장비의 로딩을 주는 것을 개선한 로딩 처리 방법에 관한 것이다.

도 1은 종래 교환기에서의 로딩을 위한 시스템 구성을 도시한 것이다.

도 1에 도시된 바와 같이, 종래 교환기에서 로딩을 위한 시스템 구성은 한 개의 시스템 관리 프로세서(11)와 하드 디스크(17), 복수개의 포트 관리 어셈블리들(12), 복수개의 메인 프로세서들(13), 복수개의 SSP(14), 복수개의 디바이스 프로세서들(15) 및 복수개의 데이터 제어 장치를 포함하여 이루어진다.

로딩 시 필요한 블록을 저장하고 있는 하드디스크 드라이브(17)와는 시스템 관리 프로세서(11)만이 연결되어 있다. 그 외의 다른 장비들은 프로세서간 통신(Inter Processor Communication, 이하 " IPC" 라 함)을 통해 하드 디스크 드라이브(17)와 연결될 수 있다. 도 1에서 실선은 로딩을 요구하는 경로를 요구한 것이다. 도 1에 도시된 바와 같이, 종래에는 로딩을 위한 장비 모두가 시스템 관리 프로세서를 통해 로딩 시 필요한 블록을 요구한다. 시스템 관리 프로세서 내에는 로딩 서버인 시스템 시작 로더가 내장되어 필요한 블록을 요구한 장비로 전송한다. 로딩이 이루어지는 과정을 자세히 살펴보면 다음과 같다.

우선 SSP(14) 또는 메인 프로세서들(13)의 부터(booter)가 시스템 관리 프로세서의 시스템 시작 로더로 프로세서 시동에 필요한 블록을 요청한다. 이를 수신한 시스템 시작 로더는 어느 장비에서 온 요청인지 판단하여 각각에 맞는 블록들을 전송한다. 이 때 메인 프로세서(13)나 SSP(14)는 프로세서에 적당한 블록인 프로그램 로더, 프로그램 로딩 데이터, OS(Operating System) 및 쉘(shell)을 수신한다. 상기한 방법으로 메인 프로세서와 SSP가 시동되면 데이터 프로세서들(15), 포트 관리 블록들(12) 및 데이터 제어 장치들(16)이 시스템 시작 로더로 로딩 요청을 하여 필요한 펌

웨어(firm ware)와 OS 응용 프로그램 블록을 로딩 받는다.

도 3은 종래 교환 시스템에서의 로딩 절차를 시스템 시작 로더와 부터별로 도시한 흐름도로서, 도 3을 참조하여 상기 로딩 과정을 상세하게 설명하면 다음과 같다.

장비 내의 부터는 시스템 관리 프로세서 내의 시스템 시작 로더로 로딩 시작 알림 메시지를 송신한다(S300). 로딩 시작 알림 메시지를 수신(S301)한 시스템 시작 로더는 메시지를 송신한 부터로 로딩 시작 알림에 대한 응답 메시지를 송신한다(S302). 로딩 시작 알림에 대한 응답 메시지를 수신한(S303) 부터는 로딩에 필요한 프로그램 블록의 헤더 정보를 요구하는 메시지를 시스템 시작 로더로 송신한다(S304). 이를 수신한 시스템 시작 로더는 프로그램 블록의 헤더 정보를 부터로 송신한다(S305). 상기 헤더 정보를 수신한 부터는 실질적인 프로그램 내용에 해당하는 프로그램 블록을 시스템 시작 로더로 요청하고(S307), 시스템 시작 로더는 이를 수신한다(S308). 시스템 시작 로더는 부터로 프로그램 블록을 송신하고(S309), 하나의 블록에 대한 로딩이 종료된 후에는 로딩 종료 메시지를 부터로 송신한다. 아직 수신할 블록이 더 남아 있을 경우에, 부터는 프로그램 블록의 헤더 정보를 요구하는 메시지를 송신하는 단계(S304)로 회귀하고, 더 이상 수신할 블록이 없는 경우에는 로딩을 종료한다.

상기한 종래의 교환기 장비 로딩 방법은 다음과 같은 문제점이 제기될 수 있다. 교환기 전체 장비가 동시에 로딩을 요청할 경우에 시스템 관리 프로세서의 부하가 증가하게 되고, 이로 인해 IPC의 유실과 명령어 처리의 지연을 가져올 수 있다. 그 결과 특정 장비의 로딩의 지연이 발생하거나 아예 로딩이 중단되는 경우가 발생하여 시스템 관리 프로세서에서 실행되는 블록이 로딩의 부하로 인해 적절한 작동을 하지 못하게 되는 경우가 있다.

발명이 이루고자 하는 기술적 과제

본 발명에서는 상기한 바와 같은 종래기술의 문제점을 해결하기 위해, SSP 내에 분산 시스템 시작 로더를 실장하여 시스템 관리 프로세서에서의 로딩 시 부하를 덜어줄 수 있는 교환기에서 분산 시스템 시작 로더를 이용한 로딩 처리 방법을 제안하고자 한다.

발명의 구성 및 작용

상기한 바와 같은 목적을 달성하기 위하여, 본 발명에 의한 교환기에서 분산 시스템 시작 로더를 이용한 로딩 처리 방법은 교환기 내 장비의 부터(booter)가 시스템 관리 프로세서에 내장된 시스템 시작 로더로 로딩 요구 메시지를 송신하는 단계(a); 상기 단계(a)의 로딩 요구 메시지를 수신한 시스템 시작 로더가 메시지를 송신한 장비가 디바이스 프로세서나 데이터 제어 장치인지 여부를 판단하는 단계(b); 상기 단계(b)에서 메시지를 송신한 장비가 디바이스 프로세서나 데이터 제어 장치일 경우, 상기 디바이스 프로세서나 데이터 제어 장치가 속하는 SSP가 현재 운용중이거나 시동중인지 여부를 판단하는 단계(c); 상기 단계(c)에서 SSP가 운용중이거나 시동중일 경우, SSP 내의 분산 시스템 시작 로더로 로딩 요구 메시지를 전달하고, 상기 분산 시스템 시작 로더는 상기 단계(a)의 로딩 요구에 대한 응답 메시지를 부터로 전송하는 단계(d); 상기 분산 시스템 시작 로더가 상기 부터로부터 프로그램 블록 헤더 정보를 요구하는 메시지를 수신하면, 상기 요구한 프로그램 블록 헤더 정보를 상기 부터로 송신하는 단계(e); 및 상기 분산 시스템 시작 로더가 상기 부터로부터 프로그램 블록을 요구하는 메시지를 수신하면, 상기 요구한 프로그램 블록을 상기 부터로 송신하고, 하나의 블록에 대한 로딩 종료 메시지를 송신하는 단계(f)를 포함하는 것을 특징으로 한다.

또한 본 발명은 상기 단계(c)에서 SSP가 운용 또는 시동중인지 여부를 판단한 후, 상기 시스템 시작 로더는 상기 로딩을 요구한 디바이스 프로세서나 데이터 제어장치로 로딩할 블록이 갱신되었는지 여부를 판단하는 단계; 및 상기 판단한 정보를 리로드(Reload) 필드로 설정하여, 상기 단계(d)에서 상기 분산 시스템 시작 로더로 전달하는 로딩 요구 메시지에 포함시키는 단계를 더 포함하는 것을 특징으로 한다.

한편, 상기 디바이스 프로세서나 데이터 제어 장치로 로딩할 블록이 갱신되었을 경우, 상기 단계(d)의 로딩 요구에 대한 응답 메시지 송신 전에, 상기 분산 시스템 시작 로더가 상기 시스템 시작로더로 갱신된 블록의 송신을 요구하고, 상기 시스템 시작 로더는 상기 분산 시스템 시작로더로 상기 요구한 갱신된 블록을 송신하는 단계를 더 포함하는 것을 특징으로 한다.

이하에서 첨부된 도면을 참조하여 본 발명에 의한 교환기에서 분산 시스템 시작 로더를 이용한 로딩 처리 방법의 바람직한 실시예를 상세하게 설명한다.

도 2는 본 발명의 바람직한 실시예에 따른 분산 시스템 시작 로더를 이용한 로딩 시스템의 구성을 도시한 것이다.

본 발명의 바람직한 실시예에 따르면 분산 시스템 시작 로더(24)는 SSP(23) 내에 실장되므로 도 2는 SSP(23)에 종속적인 디바이스 프로세서(25) 와 데이터 제어 장치(26) 및 SSP(23)에 연결된 시스템 관리 프로세서(21)에 관해서만 도시되어 있으며, 나머지 장비들은 종래와 같은 방식으로 로딩이 이루어진다.

도 2에서 실선은 교환기 내 각 장비들의 로딩이 이루어지는 경로를 도시한 것이다. 도 2에 도시된 바와 같이, SSP(23)는 종래와 마찬가지로 시스템 관리 프로세서(21)와 IPC를 통해 로딩을 하게 되나, SSP에 종속된 디바이스 프로세서(25)와 데이터 제어장치(26)는 초기의 로딩 요구는 시스템 관리 프로세서에게 하지만 그 이후의 로딩 절차는 SSP(23) 내의 분산 시스템 시작 로더를 통해 이루어진다. 단 데이터 제어 장치는 SSP가 시동되어 있지 않은 경우에도 로딩을 요구할 수 있으므로 해당 SSP가 시동되지 않은 경우에 시스템 관리 프로세서에게 직접 로딩을 요구하거나 SSP가 시동중이라면 잠시 기다린 후 SSP에 로딩을 요구한다. 도 2에는 SSP에 속하는 디바이스 프로세서와 데이터 제어장치가 하나만 도시되어 있으나 속하는 장비가 여러 개일 경우에는 여러 개의 디바이스 프로세서나 데이터 제어장치 모두가 분산 시스템 시작 로더를 통해 로딩을 수행하게 된다. 이에 대한 자세한 로딩 절차는 후술하기로 한다.

도 2에서 점선은 하드디스크 드라이브(27)에 존재하는 블록과 SSP(23) 내의 메모리(28)에 존재하는 블록과의 동기화를 위해 하드디스크 드라이브(27)에서 SSP 내의 메모리(28)로 갱신된 블록을 전송하는 경로를 도시한 것이다. 상기 동기화를 위한 자세한 방법은 후술하기로 하겠다.

도 4는 본 발명의 바람직한 실시예에 따른 시스템 시작 로더와 부터의 동작에 대한 흐름도를 도시한 것이다.

종래와 마찬가지로 장비의 부터는 시스템 시작 로더로 로딩 알림 메시지를 송신하고(S300), 시스템 관리 프로세서 내의 시스템 시작 로더는 이를 수신한다(S301). 시스템 시작 로더는 수신 후에 종래와 같이 곧바로 수신에 대한 응답을 하는 것이 아니라, 분산 시스템 시작 로더에 의해 로딩을 수행해야 하는 것인가 아니면 자신이 로딩을 수행해야하는 것인가를 판단하기 위해, 로딩 알림 메시지를 송신한 장비가 데이터 프로세서나 데이터 제어 장치인지 여부를 판단한다(S402).

로딩 알림 메시지를 송신한 장비가 디바이스 프로세서나 데이터 제어 장치가 아닌 경우(예를 들어 메인 프로세서나 SSP)에는 종래와 마찬가지로 시스템 시작 로더는 로딩 알림에 대한 응답 메시지를 부터로 송신하고(S408), 부터는 이를 수신한다(S409). 부터는 시스템 시작 로더로 직접 로딩 시 필요한 프로그램 블록의 헤더 정보를 요구하고(S410), 시스템 시작 로더는 부터로부터 요구받은 블록의 헤더 정보를 송신한다(S411). 부터는 수신한 블록의 헤더 정보를 토대로 시스템 시작 로더로 실질적인 프로그램 블록인 텍스트나 데이터 영역의 코드를 요구하는 프로그램 블록 요구 메시지를 송신한다(S413). 시스템 시작로더는 프로그램 블록 요구 메시지를 수신하고(S414), 요구받은 프로그램 블록을 부터로 전송한다(S415). 블록 전송이 완료된 후에 시스템 시작 로더는 하나의 블록에 대한 로딩이 종료되었음을 알리는 로딩 종료 메시지를 부터로 송신하고(S416), 부터는 로딩 되어야할 블록이 더 남아있는지 여부를 판단하여(S417), 로딩할 블록이 더 남아있을 경우에는 프로그램 헤더 정보를 요구하는 단계(S410)로 회귀하고, 로딩할 블록이 더 이상 없을 경우에는 로딩을 종료한다.

로딩 알림 메시지를 송신한 장비가 디바이스 프로세서나 데이터 제어 장치인 경우에는 SSP에 실장된 분산 시스템 시작 로더에 의해 로딩 작업이 수행된다. 이를 위해 시스템 시작 로더는 상기 로딩 알림 메시지를 송신한 디바이스 프로세서나 데이터 제어 장치가 교환기 내에 존재하는 여러 개의 SSP 중 어느 SSP에 속하는 장비인가를 확인한다(S403). 어느 SSP에 속하는 장비인지가 확인되면, 해당 SSP가 현재 운용중이거나 시동중인지 여부를 확인한다(S404).

해당 SSP가 현재 운용중이나 시동중이 아니라면 분산 시스템 시작 로더에 의한 로딩 작업을 수행할 수 없으므로 시스템 시작 로더에 의해 로딩이 수행되어, 시스템 시작 로더가 부터로 로딩 알림에 대한 응답 메시지를 송신하는 과정(S408)이 이루어지고, 이후에는 전술한 종래의 로딩 과정과 같은 단계로 로딩 작업이 수행된다.

해당 SSP가 운용중이거나 시동중인 경우에는 SSP 내에 실장된 분산 시스템 시작 로더에 의해 로딩이 수행된다. 도 2에 도시된 바와 같이, 분산 시스템 시작 로더를 실장한 SSP는 시스템 관리 프로서세와 같이 하드디스크 드라이브와 연결되어 있지 않고, SSP에 별도의 메모리가 존재한다. SSP에 속하는 디바이스 프로세서와 데이터 제어 장치의 로딩에 필요한 블록들은 SSP가 시스템 시작 로더를 통해 로딩을 수행할 때 시스템 시작 로더로부터 전송 받아 메모리에 저장된다. 도 2에 도시된 바와 같이, 디바이스 프로세서 로딩에 필요한 블록을 " DP_Block" 이라 하고, 데이터 제어 장치 로딩에 필요한 블록을 " DH_Block이라고 하겠다" . 일반적으로 하드디스크 드라이브에 저장된 " DP_Block" 과 " DH_Block" 은 SSP의 메모리에 저장된 " DP_Block" 과 " DH_Block" 의 내용과 같다. 그러나 경우에 따라서 하드디스크 드라이브의 " DP_Block" 및 " DH_Block" 에 대해서만 갱신이 이루어질 수 있고, 이와 같은 경우 분산 시스템 시작 로더에 의해 로딩이 이루어지면 디바이스 프로세서는 갱신이 되지 않은 SSP내 메모리의 " DP_Block" 과 " DH_Block" 을 로딩 받게 되는 문제점이 발생한다. 따라서 하드디스크 드라이브의 저장된 " DP_Block" 롸 " DH_Block" 과 메모리에 저장된 " DP_Block" 과 " DH_Block" 을 서로 일치시켜주는 동기화 과정이 필요하다.

이러한 동기화 과정은 분산 시스템 시작 로더가 디바이스 프로세서나 데이터 제어 장치로 필요한 블록을 전송하기 이전에 이루어져야 한다. 따라서 디바이스 프로세서나 데이터 제어 장치가 속하는 SSP가 운용중이거나 시동중임이 확인되면, 하드디스크 드라이브에 저장된 " DP_Block" 및 " DH_Block" 이 갱신되었는지 여부를 판단한다(S405). 본 발명의 바람직한 실시예에 따르면 갱신의 여부는 블록의 크기와 체크섬(Checksum) 정보를 확인하는 방법으로 판단한다. 시스템 시작 로더는 시동될 때마다 하드디스크 드라이브에서 데이터 프로세서와 데이터 제어 장치에 로딩될 블록의 크기와 체크섬 정보를 확인하여 시스템 관리 프로세서내 메모리에 저장한다. 후에 시스템 시작 로더가 디바이스 프로세서 또는 데이터 제어 장치로부터 로딩 알림 메시지를 송신하면, 상기 메모리에 저장된 블록과 현재 하드디스크 드라이브에 저장된 블록의 크기 및 체크섬 정보를 비교하는 방식으로 해당 블록의 갱신 여부를 판단한다. 만약 하드디스크 드라이버의 정보가 후에 갱신이 되었다면, 블록의 크기와 체크섬 정보는 서로 다르게 나타날 것이다.

하드디스크 드라이버의 블록이 갱신되지 않았다면 시스템 시작 로더는 SSP내의 분산 시스템 시작 로더로 블록이 갱신되지 않았다는 정보를 포함하여 로딩 알림 메시지를 전달한다(S407). 시스템 시작 로더에서 분산 시스템 시작 로더로 전달하는 로딩 알림 메시지는 해당 블록이 갱신되었는지 여부를 나타내기 위해 리로드(Reload) 필드를 포함한다. 하드디스크 드라이버의 블록이 갱신되었다면, 상기 리로드 필드를 " true" 로 설정하고(S406), 이를 포함한 로딩 알림 메시지를 SSP의 분산 시스템 시작 로더로 전송한다(S407).

도 5는 본 발명의 바람직한 실시예에 따른 분산 시스템 시작 로더 및 시스템 시작로더의 동작을 나타낸 흐름도이다.

상기한 바와 같이, 로딩을 요구한 장비가 디바이스 프로세서와 데이터 제어 장치일 경우, 분산 시스템 시작 로더는 시스템 시작 로더로부터 리로드 정보를 포함한 로딩 알림 메시지를 수신한다(S500). 분산 시스템 시작 로더는 로딩 알림

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.