

PC LAN을 이용한 분산 시스템의 운영방식

김 정 태 손 진 근 백 두 권

고려대학교 전산학과

< 요 약 >

본 논문에서는 PC 분산시스템의 구조와 운영방식에 대해서 논하고 자원 공유 방식과 부하 공유 방식중 어느 것을 택하느냐에 따라서 PC 분산시스템의 두가지 모델을 제시하였다. 또한 PC 분산 시스템을 위한 네트워크 운영체제의 구조, 기능 및 내재된 문제점을 알아보고 그것을 해결하기 위한 분산 운영 체제의 구조와 기능에 대해서 고찰하였다.

1. 서론

독립된 운영체제를 갖는 PC를 PC LAN으로 연결하여 구성한 분산시스템은 특수 목적의 소형 컴퓨터와 결합되어 내재적인 분산 속성을 띤다. 이러한 분산속성에는 대표적으로 자원의 분산과 각 노드의 자율성있는 제어가 있는데, 시스템을 자원공유 방식과 부하공유 방식으로 운영할 수 있다. 즉, PC 분산시스템에서는 분산된 화일, 데이터베이스, 프린터 외 주변장치, 응용소프트웨어 등의 자원을 공유하여 시스템을 운영하거나 각 노드의 자율성있는 제어를 이용하여 분산 응용소프트웨어를 병렬실행할 때 타스크 할당 기법에 의해 부하균형을 유지함으로써 운영할 수 있다. 그런데 그와 같은 분산처리 환경에서는 분산 응용 소프트웨어의 프로그래밍과 실행, 프로세스 스케줄링과 동기화, 자원할당, 분산된 정보(화일 또는 데이터베이스)의 접근 등 여러가지 분산화 문제가 발생한다.

본 논문에서는 이러한 분산화 문제를 해결할 수 있는 PC 분산시스템 운영체제를 연구하기 위하여, 네트워크 운영체제로 운영되는 PC 분산시스템이 갖는 한계와 문제점을 규명하고 이를 개선하기 위하여 분산운영체제가 갖추어야 할 구조와 기능을 고찰하고자 한다.

2. PC 분산시스템의 구성 및 운영방식

2.1 구성

다수의 PC를 PC LAN으로 연결하여 PC 분산시스템을 구성하면 통합시스템 관리체계 개발, 운영 및 유지보수 비용의 관점에서 볼 때 단일의 중앙집중형 대형시스템을 이용하는 것보다 비용효과적이다 [1].

이와 같은 동기에서 PC LAN을 이용하여 다수의 PC를 연결함으로써 구성된 분산시스템을 이 논문에서는 PC 분산시스템이라 정의한다. 다시 말하면,

PC 분산시스템이란 독립된 운영체제에 의해 정보가 이용되는 PC들 사이에 정보를 전달하고, 자원과 데이터베이스를 공동적으로 사용할 수 있도록 PC LAN을 통해 다수의 PC들을 상호접속시키므로써, 다양한 응용에 대해 전체적인 목적을 공동연산으로 수행하는 PC들의 집합체이다.

일반적인 분산시스템의 경우와 마찬가지로 PC 분산시스템에서 분산시키는 대상은 PC 하드웨어, 제어, 데이터가 있다. 이러한 관점에서 PC 분산시스템은 분산 하드웨어, 분산 제어, 분산 데이터로 구성된다 [2,3].

1) 분산 하드웨어

분산처리를 위해서는 PC 하드웨어의 물리적 분산이 선행되어야 하는데, PC 분산시스템에서는 지역 메모리와 프로세서를 갖는 PC들이 PC LAN 위에 분산된다. 또한 PC 분산시스템 내에서 상이한 기능을 갖는 PC로 이루어진 각 노드에 시스템을 기능적으로 분해하거나 응용 소프트웨어를 분산시킨다.

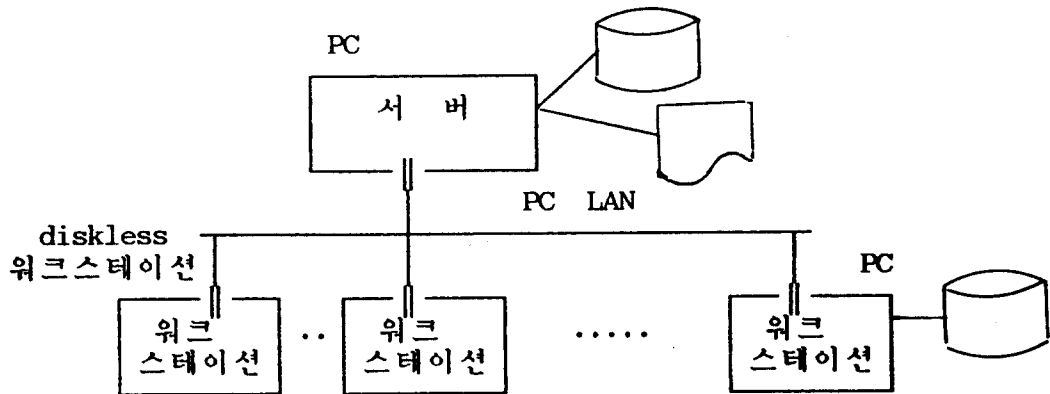


그림 1 PC 분산시스템의 물리적 구성

PC LAN을 이용한 PC 분산시스템의 물리적 구성은 그림 1과 같으며 다음과 같은 하드웨어로 이루어진다[1,4,5].

- 워크스테이션 : PC, diskless 워크스테이션 등
- 서버 : 화일서버, 프린터서버, 통신서버 등
- 네트워크 하드웨어 : 네트워크 인터페이스, 케이블, 모뎀 등
- 주변장치 : 프린터, 보조기억장치 등

2) 분산 제어

분산처리를 하기 위해서는 각 노드가 독립적으로 자원을 관리하며 업무처리를 할 수 있도록 지역자율성을 보장하여야 한다. 지역자율성은 개개의 노드에서 수행되는 동작을 조정하고 분산된 자원을 관리할 수 있도록 제어를 분산시킴으로써 얻을 수 있다. 시스템의 분산된 자원에 대해 상태정보를 일관성있게 통합, 운영함으로써 제어를 분산시킬 수 있다.

3) 분산 데이터

데이터를 분산시키는 목적은 고장에 대해 내결성을 강화하고 데이터를 생성지나 이용처에 가까운 노드에 저장하므로써 시스템의 성능을 개선하는 것에 있다. PC 분산시스템에서 데이터는 중복성과 분할을 고려하여 분산시킨다. 여러 노드에 다중의 사본을 위치시켜 중복성을 유지하고 데이터의 부분들을 다른 노드에 위치시켜서 분할시킬 수 있다.

2.2 운영방식

PC 분산시스템에 나타나는 드러진 특징은 물리적 구조의 내재적인 분산속성에 의해 일어나는 분산처리 능력과 반응시간의 향상에 있다. 그러한 특징은 PC 분산시스템을 자원공유 또는 부하공유 방식으로 운영함으로써 얻을 수 있다 [1].

1) 자원공유 방식의 운영

자원공유 방식은 PC 분산시스템의 각 노드가 갖고 있는 특정 자원들을 다른 노드가 공유할 수 있도록 하는 운영 방식이다. 서로 다른 기능 또는 자원을 갖고 있는 PC들이 연결되어 있다면 어느 한 PC에 있는 사용자는 다른 PC에 있는 유용한 자원을 이용할 수 있다.

자원공유 방식의 운영으로 원격 노드에 있는 화일의 공유, 데이터베이스의 정보처리, 원격 노드로의 화일 출력, 원격 노드에 있는 특수한 하드웨어의 사용 그리고 원격 노드에 있는 응용 프로그램의 상하로딩(up/down loading) 등을 수행할 수 있다. 원격 노드에 있는 정보를 전체 화일의 전송 또는 페이지링 기법에 의해 접근하는 데이터 이동 연산과 원격 프로시듀어 호출에 의해 접근하는 계산이동 연산 형태로 처리한다.

2) 부하공유 방식의 운영

부하공유 방식은 PC 분산시스템의 한 노드에 작업수행이 요청되고 그 작업이 동시에 실행될 수 있는 부분으로 나누어 질 수 있다면, 그 작업을 분해하여 다수의 처리요소에서 병행실행하거나, 한 노드에서 작업을 실행하고 있을 때 새로운 작업수행이 요청되면 부하가 없는 노드에 실행하도록 부하가 균형을 이루도록 하는 것이다. 부하공유 방식에서는, 작업이 한 노드에 도달했을 때 그 작업 전체 또는 일부분을 다른 노드에서 수행하게 하는 작업이동(job migration) 연산형태로 처리한다.

2.3 PC 분산시스템의 두 모델

위에서 언급한 운영방식에 따라 PC 분산시스템은 두가지 모델로 표현된다. 즉, 자원공유 방식의 운영을 위한 사용자 서버(user-server) 모델과 부하공유 방식의 운영을 위한 프로세서 풀(processor pool) 모델이 있다 [6].

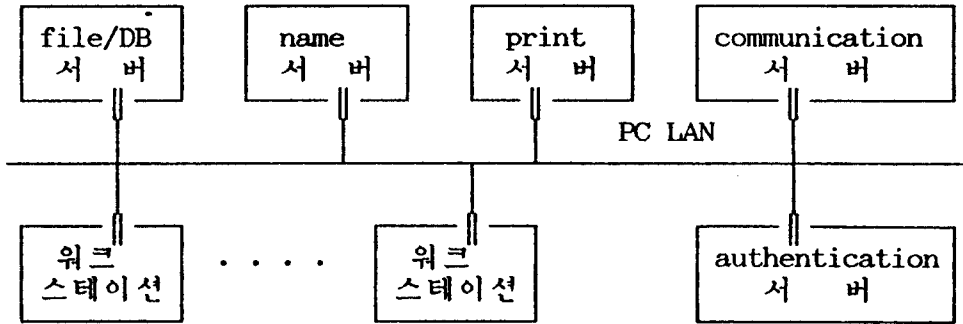
1) 사용자 서버 모델

사용자 서버 모델은 자원공유 방식의 PC 분산시스템 모델로 PC가 호스트 컴퓨터에 연결되어 지능적 단말기(intelligent terminal) 역할을 수행하는 초기모델로부터 발전한 단계이다. 이 모델의 PC 분산시스템은 기능적으로 분화된 또는 전문화된 상이한 PC들로 구성된다(그림 2의 a). 기능적으로 분화된 PC들은 프로세스간 통신 관리자, 기억장치 관리 등의 기본 서비스를 제공하는 커널 서버, 공유된 화일을 단일형식으로 접근하는 화일서버 등과 서버에 의해 관리되는 공유자원을 접근하여 사용할 수 있는 워크스테이션의 역할을 수행한다.

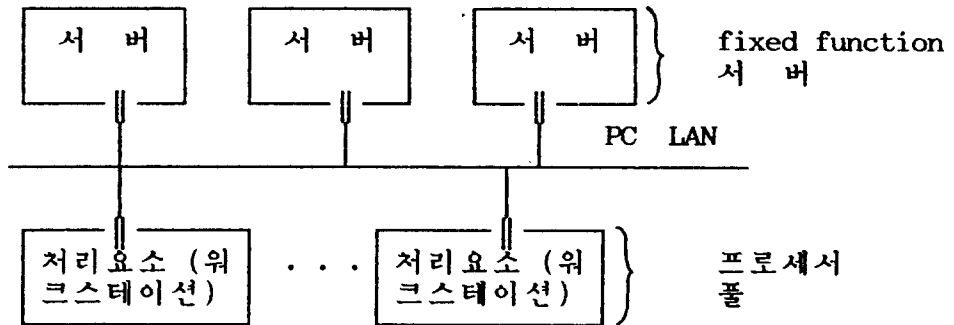
이 모델의 분산시스템은 PC의 가용성을 증가시키기 위한 동기로 구현하였으며, 각 워크스테이션은 서버로부터 공유된 자원을 이용하여 사용자에게 대화식 처리를 지원하므로써 PC의 가용성을 증가시킬 수 있다. 이때, 전통적인 운영체제에서 볼 수 있는 화일관리, 자원관리, 보호 등의 기능은 서버에서 수행되어 외부 즉, 워크스테이션에 제공된다.

2) 프로세서 풀(processor pool) 모델

프로세서 풀 모델은 부하공유 방식의 PC 분산시스템 모델로 사용자 서버 모델의 발전된 형태이다. 이 모델의 PC 분산시스템에서는 기능적으로 분화된 PC들이 서버의 역할을 수행할뿐만 아니라 작업의 기본단위를 수행할 수 있는 유사한 PC들을 이용해 프로세서 풀(processor pool)을 구성한다(그림 2의 b).



a) 사용자 서버 모델



b) 프로세서 풀 모델

그림 2 PC 분산시스템의 두 모델

사용자 서버 모델과 달리 이 모델의 PC 분산시스템에서는 하나의 노드를 처리요소로 하여 이들 처리요소들을 모아 프로세서 풀을 구성함으로써 분산처리뿐만 아니라 부하공유까지 가능하다. 작업실행이 요청되면 그 작업에 프로세서 풀에 있는 유용한 처리요소를 할당하고 작업이 끝나면 처리요소를 해제하여 프로세서 풀에 복귀시킨다. 이러한 과정에서 하나의 처리요소에서 처리가 불가능하거나 분산처리가 가능한 작업은 작업을 분해하여 다수의 처리요소에서 병행실행한다. 또한 한 노드가 작업을 수행하고 있을 때 새로운 작업수행 요청이 발생하면 부하가 없는 노드에 할당함으로써 부하를 균형있게 공유시킨다.

3. PC 분산시스템 운영을 위한 네트워크 운영체제

3.1 구조

사용자 서버모델의 PC 분산시스템은 네트워크 투명성을 제공하고 전역자원 할당 및 관리 그리고 전역 프로세스 관리를 수행하는 네트워크 운영체제에 의해 운영할 수 있다.

네트워크 운영체제는 작업에 대해서 특별히 고안된 프로그램이거나 또는 지역 운영체제에서 실행될 수 있는 타스크의 집합이며 지역 운영체제의 상위 계층에 삽입된다. 이들 타스크는 공유장치 공유기능, 공유화일 지원기능 및 기록보관, 디렉토리 관리, 사용자 관리, 프린트 스푼링, 콘솔 모니터링 등과 같은 유틸리티 기능을 수행한다.

네트워크 동작을 수행하는 이러한 타스크가 실행되기 위해서는 네트워크 운영체제의 여러 부분이 조합되어야 하는데, 서비스 요청자와 서비스 수행자가 고정되어 있고 서비스 수행자만 전역자원을 관리하므로 서비스 요청자와 서비스 수행자에 네트워크 운영체제의 필요한 모듈이 분리되어 이식된다.

네트워크 운영체제는 커널(kernel), 셸(shell), 그리고 네트워크 유틸리티로 이루어 지는데[7,8] 그 기능은 다음 절에서 설명한다. 네트워크 운영체제에 의해 운영되는 사용자 서버 모델의 PC 분산시스템은 그림 3 과 같은 계층구조를 갖는다.

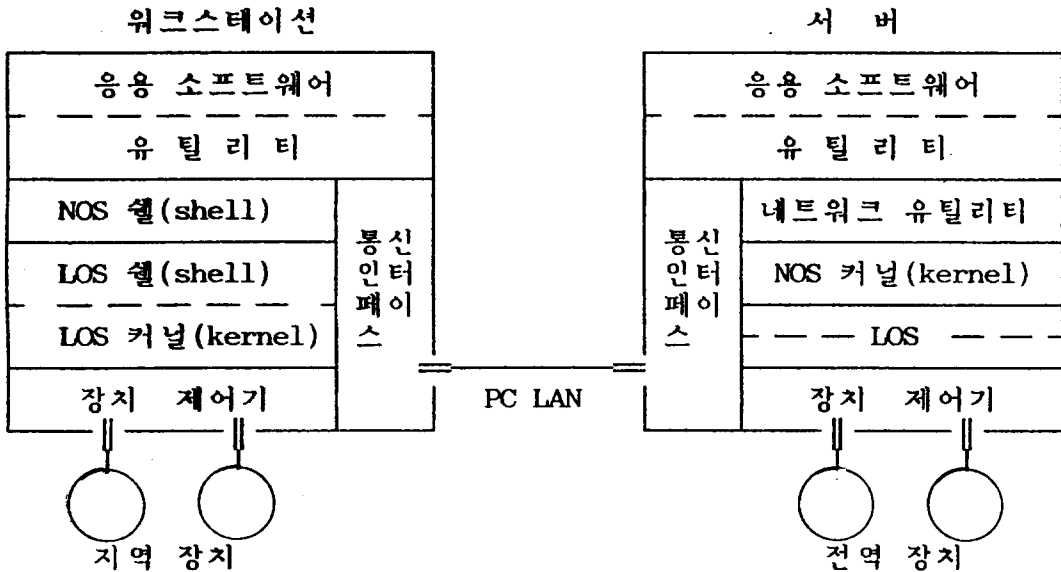


그림 3 네트워크 운영체제로 운영되는 PC 분산시스템의 계층구조

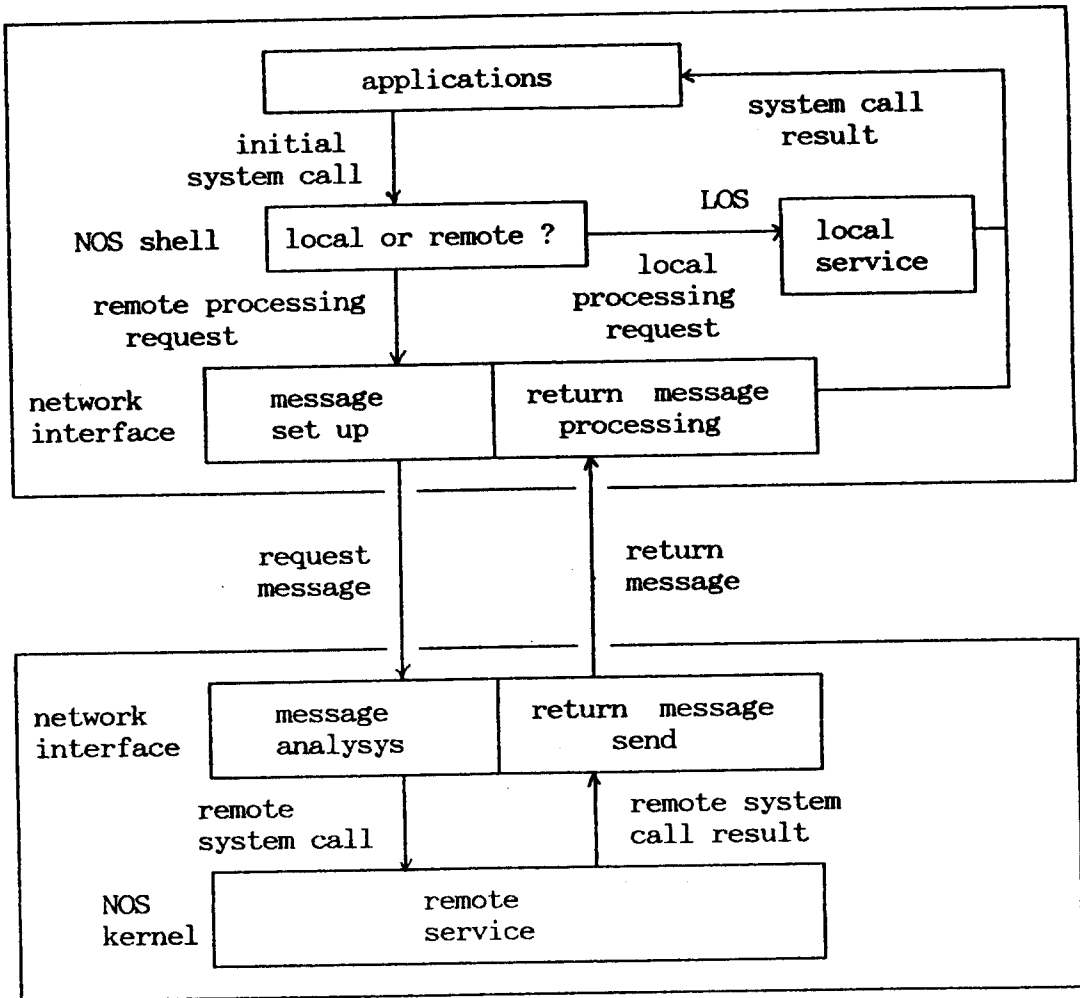
3.2 네트워크 운영체제의 기능

PC 분산시스템에서 네트워크 운영체제가 제공하는 서비스로는 화일공유 지원, 공유장치 지원, 프린트 스푼링, 네트워크 명명(naming) 서비스, 네트워크 모니터링, 액세스 제어 및 보안, 고장허용을 위한 트랜잭션 처리 등이 있다 [8].

사용자가 전역 서비스를 받기 위해서는 워크스테이션에서 네트워크

운영체제에 등록을 하고 전역 서비스 요청을 한다. 그러면, 셸은 서비스 요청이 지역적으로 처리될 것인가, 전역적으로 처리될 것인가를 분석하고 결정한다. 만일 지역적으로 수행되어야 할 서비스라면 운영체제에 의해 서비스 수행되고 전역적으로 수행되어야 할 서비스라면 네트워크 인터페이스를 통해 서버에게 전역 서비스를 요청한다. 서버는 서비스 요청이 오면 공유된 전역적 자원을 이용하여 서비스를 수행하고 수행결과를 네트워크 인터페이스를 통해 사용자에게 회신한다. 네트워크 운영체제에서 서비스가 처리되는 과정은 그림 4 와 같다.

요청자 노드 (워크스테이션)



서비스 제공 노드 (서버)

그림 4 네트워크 운영체제의 처리 흐름도

PC 분산시스템에서 서비스를 처리하는 과정에서 네트워크 운영체제의 각 부분이 수행하는 기능은 다음과 같다. 먼저, 커널은 시스템의 모든 엔티티의 상태와 위치에 대한 인지를 유지하여 원격처리 요청을 수신하였을

때 자원의 위치, 원격처리 요청에 대한 연산의 수행방법 그리고 적절한 처리결과에 대한 회신방법을 결정하는 것이 커널의 중심기능이다. 다음으로 셀은 원격처리가 필요한 네트워크 명령어에 대해 네트워크 인터페이스를 통해서 서버에게 서비스를 요청하고 서비스 수행결과를 수신하며 네트워크 명령어가 아닌 경우에는 지역 운영체제에 처리를 넘겨 지역적으로 수행한다. 또한, 네트워크 유틸리티는 네트워크 운영체제와 함께 사용되며 사용자가 네트워크 기능을 풍부하게 이용할 수 있도록 서비스를 제공한다. 기본적인 네트워크 유틸리티로는 화일 전송을 비롯하여 화일의 접근과 처리를 제어하기 위한 디렉토리 관리기능과 데이터 화일에 대한 기록보관(backup) 기능 및 화일회복 기능 등이 있다.

3.3 문제점

네트워크 운영체제에의 기능으로 해결할 수 있는 분산화 문제는 거의 없거나 미소한 영역에 불과하다. 네트워크 운영체제에서 분산처리를 할 수 없는 이유는 특히 병행실행제어, 화일 시스템 관리의 문제에서 비롯된다[3].

1) 병행실행 제어의 문제

분산 응용 소프트웨어를 연산순서 관계에 따라 분할하여 병행실행을 하는 과정은 프로세스 분할, 프로세스 스케줄링, 프로세스 할당, 프로세스간 통신 제어, 사건 스케줄링, 처리결과 병합으로 구성되는데, 이러한 과정을 조정하는 것을 병행실행 제어라 한다. 그런데, 네트워크 운영체제에서는 조정자에 의해 제어되는 조정자의 기능체와 병행실행 과정들을 수행하는 기능체가 없으므로 병행실행 제어를 할 수 없다. 따라서 분산 응용 소프트웨어의 프로그래밍과 실행, 프로세스 스케줄링 및 동기화 그리고 자원할당 문제가 해결되지 않는다.

2) 화일 시스템 관리의 문제

분산시스템에서 화일 관리 시스템 측면에서 보면 분산된 화일의 디렉토리에 대한 전역 디렉토리를 구성, 유지하여야 하며, 분산된 기억장치 모두에 대해 일관성있는 주소할당 기법을 사용함으로써 화일 접근시 모든 정보에 대한 투명성을 제공하여야 한다.

그러나 네트워크 운영체제에서는 서버에 의해 관리되는 공유화일, 공유장치 및 각 노드에 대한 위치정보만이 전역 디렉토리로 관리되므로 그외의 정보에 대해서는 투명성을 제공할 수 없다. 또한 화일의 안전성을 보장하기 위해 정보를 중복하여 분산시키게 되는데, 네트워크 운영체제에서는 화일의 중복성을 허용하지 않으므로 분산된 화일의 회복 및 고장허용의 기능을 제공하지 못한다.

4. PC 분산시스템을 위한 분산운영체제

4.1 구조

프로세서 풀 모델의 분산시스템은 부하공유 방식에 의해 분산처리가 가능한 모델로 시스템의 모든 자원에 대해 전역적 제어를 할 수 있는 분산운영체제에 의해 운영된다. 바꾸어 말하면 분산처리를 수행하기 위해서는 네트워크 투명성을 제공하는 네트워크 관리 시스템뿐만 아니라

전통적 운영체제의 모든 관리시스템들에 대해서도 가상의 단일 시스템 관점을 제공할 수 있는 분산운영체제가 필요하다[9].

PC 분산시스템에서 프로세스가 가상의 통합된 단일 시스템 관점으로 물리적, 논리적 객체와 자원을 접근하기 위해서는 명명(naming), 공유, 보호, 동기화, 프로세스간 통신, 다수의 프로세스가 접근하는 자원의 회복 등의 기능이 제공되어야 한다.

이와 같은 이유로 분산운영체제는 지역 운영체제에 소프트웨어 계층을 추가하는 것이 아니라 운영체제의 구성요소들을 전역적 관점으로 재조직하여야 한다. 따라서 분산운영체제는 네트워크 운영체제와는 달리 각 노드의 하드웨어 상위계층에 단일의 가상 시스템 관점을 갖는 운영체제로 구현된다. 분산운영체제에 의해 프로세서 풀 모델의 PC 분산시스템이 운영될 때, PC 분산시스템의 계층 구조는 그림 5 와 같이 나타난다 [2,9].

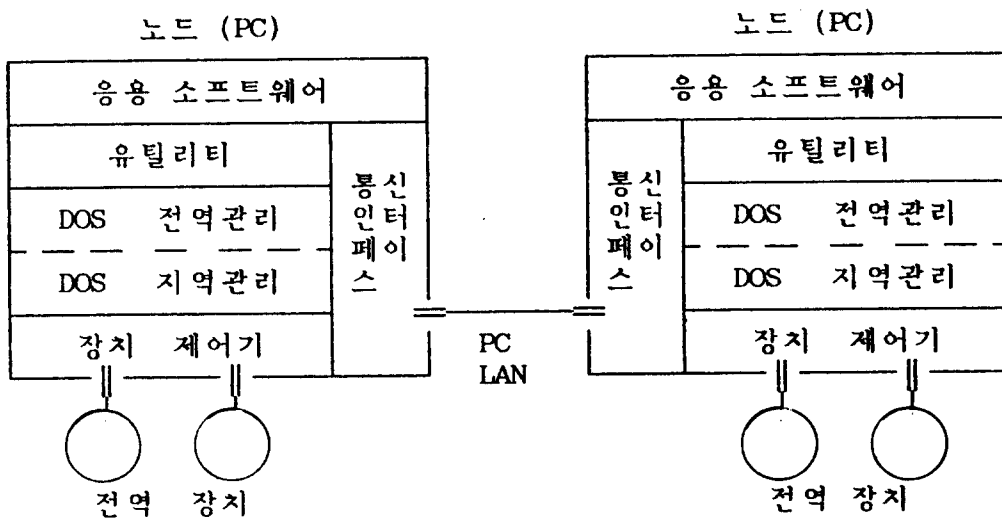


그림 5 분산운영체제로 운영되는 PC 분산시스템의 계층구조

분산운영체제의 지역관리(local management) 계층은 소프트웨어 프로세서의 생성과 제거를 관리하고 지역자원을 제어하는 시스템 소프트웨어 계층이다. 각 노드의 하드웨어를 제어하는 커널과 함께 지역자원을 관리한다. 전역관리(global management) 계층은 응용 소프트웨어나 유틸리티의 일부분으로 간주되기도 하지만 보통 몇개의 노드에서 실행되는 구분된 소프트웨어 계층이다. 전역관리 계층은 논리적으로는 하나이지만 제어는 실제적으로 여러 노드에 분산되어 운용된다.

한편, 분산운영체제를 구현하는 방법에는 시스템의 관리기능을 수행하는 기능체를 프로세스(process) 또는 객체(object) 중에 어떤 것으로 구성할 것인가에 따라서 프로세스모델 방법과 객체모델 방법이 있다.

프로세스모델 방법이란 PC 분산시스템의 자원을 관리하는 프로세스들의 집합으로 분산운영체제를 구성하는 방법이다. 이 방법으로 구현되는 분산운영체제는 프로세스 간의 통신을 관리해야 하며, 프로세스간의 동기화 및 사용자 프로세스의 제어는 프로세스간의 메시지 교환으로 해결된다. 객체모델 방법은 형태(type), 표현(representation) 및 일련의 연산(a set of operation)을 갖는 객체들의 집합으로 분산운영체제를 구성하는 방법이다. 이때는 연산을 수행하기 위해 객체에 대한

연산능력(capability)를 할당하는 과정을 통해 제어 및 동기문제를 해결한다.

4.2 기 능

객체 모델 또는 프로세스 모델에 의해 실제로 분산운영체제를 구현할 때 분산운영체제는 그림 6 과 같이 기능별로 모듈화된 소프트웨어 계층으로 구성될 수 있다 [10,11].

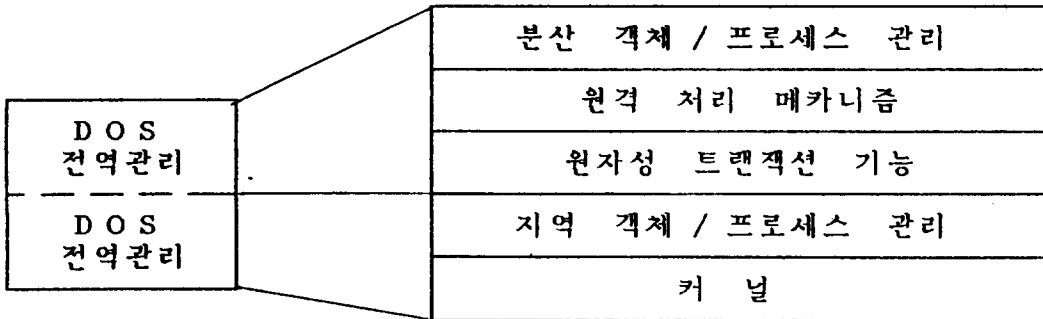


그림 6 분산운영체제의 기능

1) 커널(kernel)

커널은 PC 분산시스템의 모든 노드에서 실행되는데 각 노드의 물리적, 논리적 자원관리, 프로세스간 통신, 타스크 스케줄링, 사건 스케줄링, 인터럽트 처리 등의 기능을 수행한다. 다중 사용자의 서비스 요청을 지원할 수 있도록 다중 타스크 처리기능을 제공하고 기억장치 관리, 보호 등의 기능도 수행한다.

2) 지역 객체/프로세스 관리(local object / process management)

커널 상위에 있는 지역 객체/프로세스 관리 계층은 동시성 제어, 회복, 접근제어, 객체/프로세스 스토리지 관리 등의 기능을 제공함으로써 객체나 프로세스의 지역적 관리를 수행한다. 객체나 프로세스에 대한 지역적 관리는 객체나 프로세스의 생성과 제거를 관리하는 것을 의미한다.

3) 원자성 트랜잭션 기능(atomic transaction facility)

원자성 트랜잭션 처리기능을 제공하는 층으로 일련의 연산이 트랜잭션의 원자성을 보장할 수 있는 형식으로 수행되도록 관리한다. 즉, 이 기능은 트랜잭션의 갱신중에 고장이 발생하면 그 연산을 완료 또는 포기(all or nothing)하도록 해주는 기능으로서 신뢰성있는 연산이 수행되게 해준다.

4) 원격처리 메카니즘(remote processing mechanism)

원격처리 메카니즘 계층은 객체상에서 지역적 연산처리가 유용하지 않을 때 전역적 연산처리를 지원하기 위해 원격처리를 제어한다. 지역처리와 원격처리의 통일성을 유지하도록 메시지 전달 인터페이스가 제공되며 객체나 프로세스에 접근하는 모든 사용자에게 대해 각각의 위치 투명성을 제공하여야 한다.

5) 분산 객체/프로세스 관리(distributed object/process management)
 분산운영체제의 최상위 계층은 객체나 프로세스의 분할과 중복성을
 취급하는 분산 객체/프로세스 관리 계층이 있다. 분산된 객체나
 프로세스들의 일치성을 유지하고 동시성을 제어하는 기능을 수행한다.
 이와 같은 기능으로 구성되는 분산운영체제는 자율성이 보장된
 분산처리요소를 하나의 일관된 시스템으로 통합할 수 있어야 한다. 각
 노드에서는

- 사용자 인터페이스
 - 기능제공, 접근제어, 보호 등 시스템이용
 - 투명한 자원을 접근하는 고장허용
- 등에 관하여 공동의 제어방법이 제공되어야 한다.
 그런데, 시스템의 상태정보는 분할, 분산되어 전체 시스템 상태의
 일관성있고 완전한 관점을 형성하는 것이 불가능하다. 따라서 시스템
 전역제어는 부분적인 정보로 결정되는데 이것은 통신시스템에서의 메시지
 지연과 함께 분산운영체제를 설계하고 구성하는데 주요한 문제점으로
 남아있다.

5. 결론

본 연구에서는 PC LAN으로 연결하여 물리적으로 분산된 PC 분산시스템을
 효율적으로 운영하는 운영체제를 찾기 위하여, PC 분산시스템의 운영방식에
 대하여 알아보고 그에 따라 PC 분산시스템의 두 모델, 즉 사용자 서버
 모델과 프로세서 풀 모델을 제시하였다. 또한 네트워크 운영체제로
 운영되는 PC 분산시스템의 한계와 문제점을 살펴보고, 이러한 문제점을 할
 수 있는 PC 분산시스템의 분산운영체제의 구조와 기능에 대하여
 고찰하였다.

< 참고문헌 >

- [1] Cay Weitzman, Distributed Micro/Multi-Computer System :
 Structure, Implementation, and Applications, pp. 11-15, pp.
 220-235, Prentis-Hall, Inc., Englewood Cliffs, New Servey, 1980.
- [2] Morris Sloman and Jeff Kramer, Distributed System and Computer
 Networks, pp. 1-76, Prentis-Hall International(UK) Ltd, 1987.
- [3] 안 순신, 분산처리시스템, 정보과학회지, 제5권, 제2호, pp. 11-18,
 1987, 6.
- [4] 강 용준, LAN을 기반으로 한 작은 규모의 분산시스템, 정보과학회지
 제5권, 제2호, pp. 19-23, 1987, 6.
- [5] 김 항준, 유 기영, 분산시스템을 위한 통신망, 프로토콜 및
 화일서버의 구현, 정보과학회지, 제5권, 제2호, pp. 41-49, 1987, 6.
- [6] A. S. Tanenbaum and Rivan Renesse, Distributed Operating
 System, ACM computing surveys, Vol. 17, No. 4, pp. 419-470,
 1985, 12.
- [7] Liba Svobodova, File Servers for Network Based Distributed
 System, ACM computing surveys, Vol. 16, No. 4, 1984, 12.
- [8] 목 현상, 정 영일, LAN입문 : PC Communication & Networking,
 영진출판사, 1989.
- [9] Paul J. Fortier, Design of Distributed Operating SysTems :
 Concept and Technology, McGraw-Hill, INc., 1986.

- [10] James C. Browne, et, al., Zeus : An Object-Oriented Distributed Operating System For Reliable Applications, Proc. ACM '84 Annual Conference, pp119-187, 1984, 10.
- [11] Anard R. Tripathi, An Overview of the Nexus Distributed Operating System Design, IEEE Trans. Software Engineering Vol. 15, No. 6, pp. 686-695, 1989, 6.