

## □ 기술해설 □

## 멀티미디어 컴퓨팅을 위한 고등 운영체제

한국과학기술원 은성배\* · 김진천 · 윤천수\*\*

## ● 목

1. 서 론
2. 문제 제기
  - 2.1 대량의 데이터 복사
  - 2.2 끝-대-끝 연속성 보장
  - 2.3 시간적 동기화
3. 해결 방법들

## ● 차

- 3.1 복사방지 기법들
- 3.2 연속성 보장 기법들
- 3.3 시간적 동기 기법들
4. 시스템 예들
5. 결론 및 향후연구동향

## 1. 서 론

화상 및 음성 처리기술의 발달에 힘입어 대화형 멀티미디어 응용(Interactive Multimedia Application: IMA)이 교육용이나 상업용으로 각광을 받게 되었다. 그에 따라, IMA 자체의 개발, 운영체제, 플랫폼(platform) 등의 연구, 압축/복원이나 이미지 처리 등의 기반기술 연구 등이 학계나 업계에서 활발히 진행되고 있다. 학회활동 등을 일례로 든다면, 93년에 열린 ACM Multimedia'93 학회를 필두로 94년에는 IEEE 주최의 학회 그리고 ISMM 주최의 학회 등이 개최되었다.

멀티미디어 컴퓨팅에 관한 기존의 연구들을 학회에서 발표된 논문들을 조사하여 시기적으로 간단히 정리하면, 90년초의 논문들이 멀티미디어 응용 자체에서의 문제점을 주로 다루고 있는데 비하여 94년에 발표된 논문들은 멀티미디어 응용들을 지원하기 위하여 운영체제나 통신시스템들이 갖추어야 할 기반기능들을 주로 다루고 있는 점이 특징적이다. 이러한 시점에서 멀티미디어

어 컴퓨팅을 위한 고등 운영체제에 관한 연구들을 정리해 보는 것은 매우 시기적절하다고 할 것이다.

멀티미디어 응용은 크게 온라인(on-line) 응용과 오프라인(off-line) 응용으로 나눌 수 있다. 오프라인 응용은 그 응용내에 동영상이나 음성과 같은 연속미디어 데이터의 입력이나 출력 등의 기능을 갖지 않는 응용들로서 시간적 긴급성이 요구되지 않는다는 특징이 있다. 예를 들면, 멀티미디어의 저작과 같은 응용이나 정지영상의 저장/검색 등이 그 예이다. 반대로, 온라인 응용은 그 응용내에 연속미디어 데이터의 입력, 혹은 출력을 갖는데 연속미디어의 입출력에 요구되는 연속성이나, 동기화 등의 시간적 제약을 유지해주는 것이 문제가 된다. 예를 들어, 비디오나 오디오 정보의 검색 시스템, 원격영상회의 시스템 등에서는 데이터가 사용자의 출력 디바이스에 시간적 제약이 유지되면서 출력되어야 한다. 오프라인 응용들을 위해서는 시분할형의 운영체제로써 충분히 지원할 수 있으나, 온라인 응용을 위해서는 그 응용들에 내재된 시간적 제약을 지켜줄 수 있도록 실시간 스케줄링[1] 등의 기능을 운영체제가 지원해야 한다.

\*정회원

\*\*중신회원

멀티미디어 컴퓨팅을 위한 운영체제에서 문제가 되는 것들을 크게 세가지로 정리할 수가 있다. 첫째는 비디오나 이미지 등의 디지털 데이터들의 양이 매우 크다는 것이다. 이 때문에 UNIX 등의 운영체제에서 사용되는 입출력 기법에서는 대량의 데이터가 복사되는 문제점이 발생한다[2,3,4]. 둘째는 비디오나 음성과 같은 연속미디어(continuous media)들이 일정한 속도로 처리되거나 출력되어야 한다는 것이다. 예를 들어, 서버에 저장되어 있는 비디오를 온라인으로 전송받아 화면에 출력하는 응용을 생각해 보면, 일초에 30 프레임의 비디오 데이터가 출력장치에 끊임없이 공급되어야 한다. 이렇게 연속적인 데이터 공급을 위해서 서버의 화일 시스템, 중앙처리기, 통신시스템, 각각에서 비디오 데이터가 연속적으로 처리되어야 한다. 이러한 문제를 끝-대-끝(end-to-end) 연속성 보장문제라 한다[3,5]. 끝으로, 비디오나 오디오, 텍스트 등의 미디어들이 복합(composition)되어 출력될 때, 각각의 미디어들을 동시에 맞게 출력하는 문제[6,7]를 들 수 있다.

본 기고에서는 각각의 문제들을 상세히 알아보고 이를 해결하기 위한 기존의 연구들을 조사한다. 또한, 지금까지 제안된 멀티미디어 컴퓨팅을 지원하는 운영체제들을 소개하고 이들의 장단점을 기술한다. 본 기고의 구성은 다음과 같다. 2장에서 멀티미디어 컴퓨팅을 위한 운영체제의 문제점을 기술하고, 3장에서 이의 해결을 위한 기존의 연구들을 소개한다. 4장에서 지금까지 제안된 운영체제의 예들을 소개하고, 5장에서 결론 및 향후연구방향을 기술한다.

## 2. 문제 제기

이 절에서는 멀티미디어를 위한 운영체제에서 해결해야 할 문제점들을 기술한다.

### 2.1 대량의 데이터 복사

멀티미디어 컴퓨팅의 특징중의 하나는 처리해야 할 데이터의 양이 매우 많다는 것이다. 예를 들어 NTSC 방식의 동영상의 경우에 512×480의 해상도, 각 픽셀당 3색, 초당 30 프레임의

구성을 갖는데, 이는 22.1 Mbytes/sec의 데이터에 해당한다. MPEG1으로 압축된 상태에서도 연속적으로 1.5 Mbits/sec를 처리해야 한다.

기존의 운영체제에서 이러한 대량의 데이터를 처리할 때의 문제점은 데이터의 입출력시에 입출력 디바이스가 사용하는 버퍼와 사용자 프로세스의 버퍼 사이에 대량의 데이터 복사가 불필요하게 일어난다는 점이다. 예를 들어, 화일에서 비디오 데이터를 단순히 읽어서 화면에 뿌려주는 응용은 그림 1에서 볼 수 있는 것처럼 read후에 이를 write하는 형태로 프로그램된다. 이때, 그림 1은 UNIX에서 사용하는 입출력 버퍼의 구조를 보여주고 있는데, 그림에서 볼 수 있는 것처럼, 사용자의 요구에 따라 입력 디바이스는 디바이스 드라이버의 버퍼에 지정된 데이터를 저장하고 처리를 끝낸다. 이 데이터는 그 프로세스가 깨어나서 입력 명령의 뒤부분을 처리하면서 사용자 프로세스내의 버퍼로 복사가 되는데, 기존의 범용 운영체제에서는 처리되는 데이터의 양이 크지 않으므로 크게 문제되지 않으나 대량의 데이터를 처리해야 하는 멀티미디어 운영체제에서는 큰 문제가 되고 있다.

### 2.2 끝-대-끝 연속성 보장

멀티미디어 중에서 비디오나 음성과 같이 연

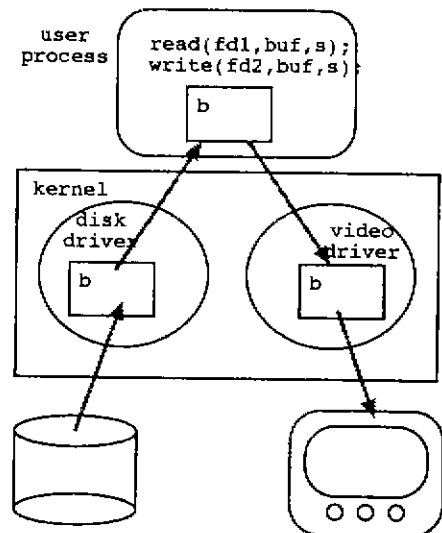


그림 1 버퍼간의 대량의 데이터 복사

속성을 갖는 미디어를 연속미디어(continuous media)라고 한다. 이러한 연속미디어는 연속성을 유지하기 위하여 운영체제의 특별한 지원을 받아야 한다. 예를 들어, 대화형 멀티미디어 응용의 일례로서 요즘 한창 각광을 받고 있는 VOD (Video On Demand) 응용을 생각해 보자. VOD 응용은 비디오 대여점에 가서 비디오를 빌리는 것이 아니라 가정의 TV 상에서 비디오를 고르고, 그 비디오가 통신망을 통하여 가정의 TV로 직접 전송되는 형태의 응용이다. 이때, 비디오 데이터를 모두 받은 후에 그 비디오를 오프라인으로 출력하는 것이 아니고 비디오 데이터가 전송되는 데로 온라인으로 TV에 출력된다.

이러한 응용에서 문제가 되는 것은 비디오 데이터가 일정한 비율로 연속적으로 비디오 서버로부터 가정의 출력 장치까지 전송되어야 한다는 것이다. 예를 들어 MPEG1[8]의 경우엔 1초에 15 Mb/s의 데이터가 전송되어야 하는데 만약, 전송률이 그 이하가 되면, 출력장치의 버퍼에는 데이터가 부족하게 되며 전송률이 그보다 크면, 출력 장치의 버퍼가 넘치게 된다. 또한, 평균적인 전송률은 유지하지만 순간순간의 부하에 따라 전송률이 크게 변하면 마찬가지로 버퍼의 한계를 넘어 버릴 수 있다.

그림 2는 VOD 응용과 같이 비디오 데이터가 서버로부터 사용자에게 전송될 때에 서버내에서 연속성을 보장하기 위한 단계를 보여주고 있다. 그림에서 볼 수 있는 것처럼, 디스크의 물리적 불력에 화일 시스템의 논리적 불력을 할당하는 단계에서 부터 연속성 문제는 시작한다. UNIX와 같은 기존의 운영체제[9,10]에서는 디스크내에서 단편(fragmentation)에 의한 낭비를 막기 위하여 불력을 임의로 할당하는 정책을 쓰고 있는데, 이 정책은 연속미디어를 접근할 때에 연속성을 저해하는 작용을 한다[5]. 다음으로는 디스크의 접근이 동시에 요구될 때 디스크 헤드의 위치에 따라 이들 요구중에서 어떤 요구를 먼저 처리할 것인가를 결정하는 것이 문제가 된다. 다음 단계는 화일 시스템에 들어오는 요구들 중 어떤 요구를 먼저 처리할 것인가이며, 그 다음 단계에서는 CPU에서 여러개의 연속미디어를 처리할 때 어떤 미디어 스트림을 어떻게

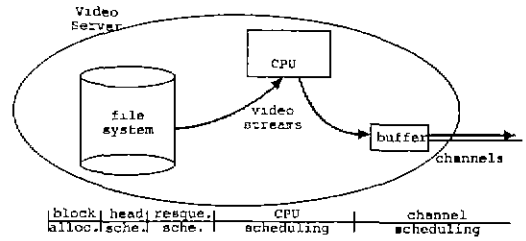


그림 2 연속미디어 전송에서 연속성 보장 단계

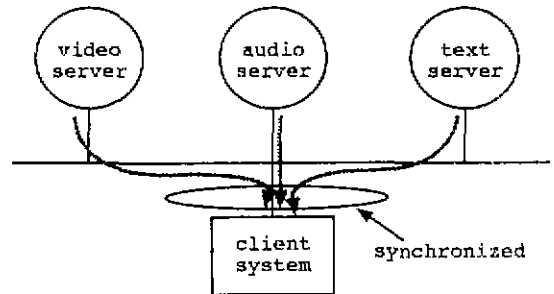


그림 3 분산 서버 환경에서의 시간적 동기화

처리할 것인가가 문제가 된다. 끝으로, 다양한 형태의 통신망에서 연속미디어의 시간제약을 어떻게 지원할 것인가가 문제가 된다. 이상과 같은 여러 단계에서 연속미디어의 연속성 문제가 여러 형태의 실시간 스케줄링 문제[1]로 귀결되며, 많은 연구에서 이러한 연속성 문제를 실시간 스케줄링에 의해 풀고 있다.

### 2.3 시간적 동기화

비디오나 음성 등의 단일미디어(monomedia) 내에서 연속성이라는 시간적 제약이 있으며, 단일미디어 간에도 시간적 동기화라는 시간제약이 존재한다. 이러한 미디어간의 시간적 동기는 텍스트나 이미지와 같은 비연속미디어(noncontinuous media)에서도 발생하기 때문에 더욱 복잡해진다. 그림 3에서 볼 수 있는 것처럼, 각각의 모노미디어들이 각각의 분산된 서버에 저장되어 있고, 이들을 클라이언트에서 실시간에 조합(composition)하여 검색하는 응용[7]을 그 예로 생각할 수 있는데, 이러한 시스템에서 데이터를 온라인으로 검색하려면, 각각의 서버에 검색을

구를 보내고 각각의 서버들이 서로 시간적으로 동기된 시점에 검색된 결과를 클라이언트에게 전송해야 할 것이다. 이때, 서버들이 시간제약을 책임질 것인가, 혹은 클라이언트만이 시간제약을 책임질 것인가? 또는, 광역(global) 시간을 가정할 것인가, 말 것인가 등, 다양한 종류의 해결책을 생각할 수 있다.

### 3. 해결 방법들

이 절에서는 멀티미디어 컴퓨팅을 위하여 운영체제를 개선시키려는 관련연구들을 살펴본다.

#### 3.1 복사방지 기법들

대부분의 운영체제에서 채택되고 있는 read/write에 의한 입출력 프리미티브에서는 디바이스 드라이버와 사용자 프로세스 간의 대량의 데이터 복사가 불필요하게 일어난다. 따라서, 이 문제를 해결하려는 연구들은 디바이스 드라이버간에 직접 데이터 복사가 가능하도록 하는 방법들을 고안하고 있는데 기반이 되는 운영체제에 따라서 다른 구조를 보이고 있다. Nakajima[2]는 Mach 운영체제에서 사용자 정의 디바이스 드라이버를 정의할 수 있게 함으로써 문제를 해결하고 있다. 즉, 디바이스 드라이버를 사용자가 직접 정의해서 쓸 수 있으므로 사용자 프로세스에 데이터 복사가 불필요한 경우엔 다른 디바이스로 데이터를 직접 복사하도록 구현하면 된다. 이러한 해결책은 Mach가 시스템 레벨에서 최소의 커널만을 가지고 있고 화일 시스템, 통신시스템 등의 대부분의 시스템 프로그램을 사용자 레벨에서 정의하고 있으므로 매우 자연스럽다.

Fall[4]은 UNIX 시스템에서도 동일한 문제가 발생한다는 점을 지적하고, 디바이스 드라이버와 드라이버간의 직접 연결을 지원하는 "splice"라는 시스템 명령어를 제안하였다. 시스템 명령어의 구조는 "splice(fd1, fd2, size)"의 형태로서, 화일 fd1과 fd2 사이에 "size" 크기의 splice를 정의하고, fd1에 데이터가 들어오면, 자동으로 fd2로 자료가 복사되도록 한다. 이 splice를 이용하여 그림 4 처럼, 네트워크 드라이버, 압축/복원

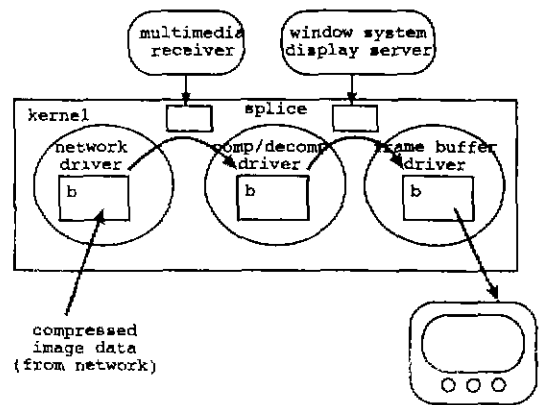


그림 4 Splice의 응용 예

드라이버, 그리고 후레임 버퍼(frame buffer) 드라이버 간에 두번의 자료 복사만으로 데이터를 처리할 수 있음을 보였다. splice 없이 사용자 프로세스가 중계하는 경우에는 4번의 자료 복사가 필요하다. splice를 UNIX에 구현하여 성능평가를 해 본 결과 55%의 성능 향상을 얻었다고 한다.

AVK(Audio Video Kernel)[11]는 기존의 운영체제에 멀티미디어 컴퓨팅을 위한 기능을 첨가한 것이 아니고 새로운 개념 모델을 바탕으로 설계, 구현된 작은 운영체제이다. AVK는 Intel의 DVI 보드위에서 사용자에게 비디오 입력, 오디오 입력, 비디오/오디오 믹싱, 비디오 압축/복원, 오디오 압축/복원, 그리고 이들 기능의 결합 등의 다양한 기능을 지원한다. AVK의 기능들을 이용하여 사용자 프로세스는 자신의 프로세스로 데이터를 복사하는 일없이 데이터를 처리할 수 있다. DVI 보드의 초기버전에는 ActionMedia750 [12]이라는 소프트웨어가 제공되었는데, ActionMedia750은 SuperVCR 모델에 기반을 두고 있어서 비디오를 비디오 플레이어에서 다루는 것처럼, 녹화, 재생, 전진, 후퇴 등의 기능만을 지원하는데 반하여 AVK는 Digital Production Studio의 개념모델을 바탕으로 비디오 및 오디오의 편집, 믹싱, 그리고 이미지 처리 등의 기능을 추가로 지원하고 있다.

#### 3.2 연속성 보장 기법들

연속미디어를 처리하기 위해서는 시스템의 각 단계에서 연속성을 고려한 처리기법들을 적용해야 한다. 이러한 연속성 보장 기법의 핵심은 다양한 서비스 요구들의 시간제약들을 미리 검토하여 스케줄하는 데에 있으며, 스케줄이 어려운 경우에는 서비스의 요구를 거부하게 하는 승인 제어(admission control) 기법도 중요한 요소이다.

### 3.2.1 블럭 할당 정책

블럭 할당 문제는 디스크에 물리적 블럭을 할당할 때에 연속미디어의 연속성을 고려하여 할당해야 한다는 것이다. 이 블럭 할당 문제는 디스크의 쓰기가 가능한지의 여부에 따라 정적 할당과 동적 할당의 두가지 문제로 나누어진다. 먼저, 정적 할당은 광디스크와 같은 읽기전용 디바이스에서 비디오나 오디오와 같은 연속미디어를 할당하는 문제이다. 이때, 광디스크에 하나의 연속미디어만을 저장하는 경우에는 단순히 비디오 스트림의 연속된 논리적 블럭을 연속된 물리적 블럭에 할당하면 되나, 비디오와 오디오가 함께 동기화되어 할당되려면, 비디오 데이터와 오디오 데이터를 섞어서 할당해야 한다. Wells [13]는 두 미디어의 블럭을 섞을때 단편(fragmentation)을 최소로 하면서 두 미디어간의 시간적 동기를 보장할 수 있는 최적의 할당 알고리즘을 제안하였다.

쓰기가 가능한 하드디스크에서의 블럭할당 문제는 광디스크에서 보다 복잡하다. 예를 들어, 기존의 UNIX[9]와 같은 운영체제에서는 식재와 쓰기가 계속될 때 발생하는 단편(fragmentation)을 줄이기 위하여 자유 할당(random allocation) 정책을 쓰고 있는데 연속미디어처럼 논리적으로 연속된 블럭을 검색할 경우에는 검색시간이 비예측적(unpredictable)이 되므로 적합하지 않다. 또 다른 문제는 다수의 동기화된 연속미디어를 함께 접근할 때에 연속적인 접근이 가능하도록 할당해야 한다는 것이다. Venkat[5]은 미디어들을 하드디스크에 나선모양(spiral)으로 할당하는 기법을 제안하고 성능을 분석하였다.

### 3.2.2 디스크 헤드 스케줄링

디스크 접근시에 가장 시간이 많이 걸리는 부분은 디스크헤드가 요구된 트랙으로 찾아가는 탐색시간(seek time)이다. 따라서, 디스크에서 다수의 접근요구를 처리할 때 어떤 요구를 먼저 처리할 것인가가 문제가 되는데 탐색시간을 최소로 줄이면서 다수의 요구를 서비스하는 SCAN, 또는 F-SCAN과 같은 알고리즘들이 제안되었다 [10]. 그러나 기존의 알고리즘에서는 연속미디어의 연속성을 고려하지 않기 때문에 각각의 미디어의 연속성이 보장되지 않는다.

Reddy[14]는 SCAN 알고리즘의 문제점을 보완하기 위하여 실시간 스케줄링 기법으로 널리 알려진 EDF(Earliest Deadline First) 알고리즘을 헤드 스케줄링에 도입한 SCAN-EDF 알고리즘을 제안하였다. SCAN-EDF 알고리즘은 먼저 각 요구의 마감시간을 조사하여 마감시간이 가장 빠른 요구를 선택한후, 마감시간이 동일한 요구에 대해서는 헤드의 이동방향쪽으로 최초인 요구를 서비스한다. 또한, 승인제어 조건으로서 새로운 요구의 주기가 현재 서비스되고 있는 요구들의 총서비스 시간보다 크기만 하면 된다는 것을 보였다.

Chen[15]은 기존의 고정순서(fixed order) 알고리즘이나, SCAN 알고리즘을 확장한 Grouped Sweeping Scheduling(GSS) 알고리즘을 제안하였으며, 요구되는 미디어의 형태가 동일한 경우, 즉 주기가 같은 경우에, GSS가 최적이라는 것을 보였다. 또한, 요구되는 미디어의 형태가 상이할 경우, GSS를 요구 미디어의 형태에 최적화시킬 수 있음을 보였다.

Vin[16]은 디스크 헤드 스케줄링 문제를 그래프의 순회문제로 변환하여 풀려고 하였다. 즉, 디스크 접근 요구들로부터 정점은 요구한 데이터 블럭을 나타내고 간선은 데이터 블럭간의 물리적 거리를 나타내는 완전연결지향(fully connected directed) 그래프를 만들고 헤드스케줄링 문제를 이 그래프의 모든 노드를 최소의 비용으로 순회하는 TSP(Traveling Sales Problem) 문제로 변환하였다. 이때, TSP 문제가 NP-complete[17] 문제이므로 휴리스틱 알고리즘을 제안하였다.

이상에서 살펴본 것처럼, 디스크 헤드 스케줄링을 위하여 다양한 알고리즘이 제안되고 있으

며, 이들간의 성능평가가 중요한 문제로 부각되고 있다. Lim[18]은 시뮬레이션을 통하여, FCFS(First Come First Served), SCAN 등의 비실시간 스케줄링 기법들이 예상대로 시간제약을 잘 지키지 못함을 보였다. 또한, 기존의 실시간 스케줄링 기법들인 EDF, D\_SCAN(Earliest Deadline SCAN), P\_SCAN(Priority SCAN), SSEDV(Shortest Seek and Earliest Deadline by Value)[19] 등을 시뮬레이션하여 그중에서 EDF와 SSEDV가 마감시간안에 데이터 접근을 가장 잘 지원하는 것을 보였다.

### 3.2.3 디스크 요구 스케줄링

VOD 응용에서는 많은 수의 비디오를 저장하고 있다가 사용자의 요구에 따라 디스크에서 검색하여 이를 가입자 선로에 실어주게 된다. 이러한 시스템에서 사용자 검색 요구의 특징은 검색되는 연속미디어들이 검색주기가 같거나, 전송되는 데이터의 양이 같은, 동형(homogeneous)의 요구들이라는 것이다. 이러한 특정화된 시스템에서는 디스크 헤드 스케줄링, CPU 스케줄링, 통신 채널 스케줄링 등의 단계를 거치지 않고 화일 시스템에서 사용자요구의 스케줄링만으로 처리가 가능하고, 동형의 요구들이므로 round-robin과 같은 간단한 스케줄링만으로 연속성 보장이 가능하다. 미네소타 대학에서는 VOD 서버를 설계하면서 round-robin을 이용하여 화일 시스템에서 사용자요구의 스케줄링하는 기법을 제안하였다[20]. 사용자 요구들은 하나의 round에서 모두 서비스를 받게 된다.

### 3.2.4 처리기 스케줄링

처리기에서 연속미디어를 다루는 문제는 하나의 처리기에서 보통 비연속미디어의 처리, 사용자 프로그램의 수행, 시스템 관리 프로그램의 수행 등, 연속미디어와 관계없는 예측하기 어려운 수행들을 함께 처리하기 때문에 디스크 헤드 스케줄링 문제 등과 달리 단순하지 않다. 그렇다고 해서 연속미디어를 위한 배려를 하지 않으면 연속미디어들의 연속성은 보장될 수가 없다. 이러한 문제를 해결하기 위하여 Mercer[21]는 시분할시스템의 운영구조를 유지하면서 연속미

디어의 연속성을 지원하는 방안으로 연속미디어의 처리에 필요한 시간을 미리 확보하는 방안을 제안하였다. 처리기에서의 스케줄링 기법은 고정우선순위(fixed priority) 스케줄링이나 가변우선순위(dynamic priority) 스케줄링을 채택하였고, 각각의 스케줄링 기법에 대하여 승인제어 조건들을 제안하였다. 이상의 스케줄링 기법을 Real-Time Mach에 구현하여 실험한 결과 예측 가능한 성능을 얻을 수 있음을 보였다.

이와 달리 일반적인 운영체제에서 연속미디어를 다루는 것이 아니라 AVK[11]와 같은 특정 응용을 위한 운영체제를 설계할 경우에는 요구들의 특성을 규정하는 파라미터가 잘 알려지게 되므로 좀 더 정확한 스케줄링을 통하여 자원의 낭비를 막을 수 있다. 이러한 면에서 Mei[22]는 특정 멀티미디어 응용을 분산처리기에서 수행시키는 문제를 제안하였다. 즉, 멀티미디어 응용을 각각의 정점이 연속미디어의 생성(source), 소멸(sink), 처리(process)를 나타내고 그들간의 간선이 연속미디어의 흐름을 나타내는 지향(directed) 그래프로 표현하고 각각의 간선에는 연속미디어의 흐름의 양을 나타내는 무게(weight)를 부여하였다. 또한, 이 그래프를 분산처리기에 할당하는 알고리즘을 제안하였다. Ryu[23]는 CPU에서 연속미디어를 처리할 때 스케줄 부하나 인터럽트 처리 부하 등의 비예측적인 이벤트를 처리하기 위하여 서버프로세스를 두는 기법을 제안하였다. 이 프로세스는 주기적으로 스케줄되고 가장 높은 우선순위를 갖는데, 수행가능조건의 분석을 통하여 그 프로세스의 수행시간을 결정할 수 있다.

### 3.2.5 통신채널 스케줄링

통신채널에서의 스케줄링도 처리기 스케줄링에서와 마찬가지로 두가지 접근방법이 있다. 그중 하나는 채널에서 전송해야 하는 데이터의 시간특성을 입력받아서 전송중의 연속성을 보장하는 기법이고 다른 하나는 시스템에서 전송되는 데이터의 특성을 알기 어려울 때 우선순위에 기반을 두고 전송을 하는 기법이다.

각각의 경우를 살펴보면 먼저 전송될 데이터의 특성을 잘 알고 있을때 Zheng[24]은 일반적인 점-대-점 통신망에서 주기와 통신데이터의 양,

그리고 그 데이터의 마감시간 등의 파라미터가 주어졌을때 마감시간을 넘기지 않는 것을 보장하는 선점(preemptive) 스케줄링 기법과 그의 승인제어 조건을 제시하였다. 스케줄링 기법으로는 EDF 알고리즘을 사용하였으며, 승인제어 조건은 의사다항(pseudo-polynomial) 시간을 갖는 것으로 분석되었다. 또한, Zheng은 통신망에서 선점(preemptive) 스케줄링을 구현하는 것이 어렵다는 점을 지적하고 비선점 EDF 알고리즘에서의 승인제어 조건도 제안하였다. 그의 연구는 Jeffay[25]의 비선점 EDF 알고리즘에서의 스케줄 가능성 검사조건과도 매우 유사하다. 점-대-점의 승인제어 조건을 바탕으로 인접되어 있지 않은 노드들간에 채널을 설정하는 알고리즘도 제안하였다. Baek[26]도 Zheng과 마찬가지로 비선점 EDF 알고리즘을 기반으로 멀티미디어 데이터를 스케줄링하는 기법을 제안하였다. 그의 접근법의 특징은 멀티미디어 데이터 전송요구들이 일반적인 실시간 통신 요구와 달리 동형의 요구가 많다는 점에 착안하여 모든 요구가 주기가 같거나 혹은 모든 요구의 전송부하가 같다는 가정하에서 승인제어 조건을 제시한 것이다. 만약, 주거나 전송부하가 다를 경우에는 버퍼의 크기 등을 변화하여 동일하게 바꿀 수 있다는 것을 보였다.

전송될 데이터의 특성을 잘 알지 못하는 경우에는 우선순위를 기반으로 스케줄링하는 고정 우선순위 스케줄링 기법을 고려할 수 있다. 이런 기법에서는 먼저 데이터 요구의 특성에 따라 각각의 요구에 우선순위를 부여하고, 우선순위가 높은 데이터가 도착하면 이미 도착되어 있는 전송 요구들보다 먼저 스케줄링을 한다.

두 기법과는 별개로 전송시에 시간제약이 지켜지지 않는 경우에 사람의 눈이나 귀가 시각정보나, 청각정보의 변화에 둔감하다는 특성을 이용하여 전송되는 데이터의 양을 줄여서 시간제약을 지키려는 기법들도 소개되어 있다. Park [27]은 오디오 폰을 컴퓨터 망에서 구현할 때에 통신채널의 지연때문에 연속성이 깨어지는 경우, 오디오 데이터를 쪼개서 시간제약을 지키려는 기법을 사용하였다. 마찬가지로, Anderson[28]도 분산 멀티미디어 응용에서 시간제약을 어기게

되는 경우, 전송되는 데이터의 중간, 중간을 버리고 나머지만을 전송하는 기법을 사용하였다. Chakrabarti[29]는 비디오 전송시에 압축과 복원이 온라인으로 수행되는 시스템에서 비디오의 압축률을 통신망의 부하에 적응적으로 조정하여 시간 제약을 지키는 기법을 제안하였다.

### 3.3 시간적 동기 기법들

VOD 응용의 경우에는 하나의 화일에 영화의 펄스와 마찬가지로 비디오와 오디오가 함께 혼합(interleaved)되어 있다. 이렇게 혼합된 데이터 구조에서는 동기화가 유지되어 있으므로 통신망에서 전송시에 미디어간의 동기화 문제가 일어나지 않는다. 그러나, 그림 5-1)에서 처럼 비디오와 오디오가 서로 다른 서버에 저장되어 있다 가 클라이언트의 요구에 의해 실시간에 조합되어 출력될 경우에는 각각의 서버에서 비디오와 오디오가 독립적으로 클라이언트에 도착하므로 통신채널간의 시간적 지연의 차이에 따라 동기가 어긋난다는 문제가 발생한다. 단일 서버의 경우에도 비디오와 오디오가 혼합되어 있지 않고 서로 다른 화일에 저장되어 있다면, 그림 5-2) 처럼, 각각의 미디어들이 서로 다른 채널로 전송되어야 하는데 각각의 채널이 서로 다른 통신 지연 특성을 보이므로 클라이언트에서 이들의 시간적 동기를 유지시켜 주어야 한다.

시간적 동기 기법의 관점에서 두 시스템은 서로 다른 특징을 보이는데 먼저 1) 과 같은 분

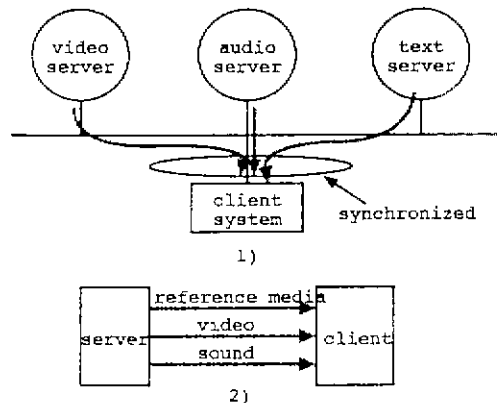


그림 5 서버/클라이언트 시스템에서의 시간적 동기화

산서버의 경우에는 서버간에 동기를 위한 노력이 큰 부담이 되고, 시스템내에 광역 클럭을 상징하기도 어렵다는 특징이 있다. 2)의 경우에는 단일 서버이므로 서버간의 동기화가 필요없고, 전송되는 미디어 중의 하나를 참조(reference) 미디어로 하여 클라이언트에서 동기를 유지할 때 도움을 줄 수 있다.

Lamont[71]는 멀티미디어 뉴스 응용 시스템에서 뉴스의 각 요소, 즉, 비디오, 아나운서의 음성, 자료 텍스트 등이 각각의 서버에 저장되어 있는 경우에 클라이언트가 이들 독립된 미디어들의 시간적 동기를 유지하는 SSP(stream synchronization protocol) 프로토콜을 제안하였다. SSP는 순수한 소프트웨어에 의해 제어되는 프로토콜로서 광역 클럭을 가정하지 않고 동기화에 서버의 도움도 받지 않는다는 특징을 갖는다.

Kim[30]은 원격영상회의 시스템에서 실시간에 입력되는 비디오나 오디오와 같은 연속미디어들을 다중채널로 전송할 때의 동기화 문제를 미디어내 동기화(intra-media synchronization)와 미디어간 동기화(inter-media synchronization)로 나누고 각각의 경우에 해결책을 제안하였다. 미디어내 동기화 문제는 연속미디어의 연속성을 보장하는 것을 의미하는 데, 확률모델을 기반으로 버퍼의 양을 결정하는 정책을 제시하였다. 미디어간 동기화는 서버가 제공하는 참조 미디어를 바탕으로 클라이언트가 다른 미디어와의 동기를 유지하도록 하였다.

#### 4. 시스템 예들

지금까지 운영체제의 각 부분에서 멀티미디어 컴퓨팅을 위하여 지원되어야 하는 기능들을 조사하고 관련 연구들을 살펴보았다. 이러한 각 부분들을 조합하여 하나의 운영체제로 구현할 때에 크게 두 그룹으로 나눌 수 있는데 멀티미디어 컴퓨팅에서 요구되는 실시간 특성들을 미리 추출하여 정적으로 스케줄하는 정적(static) 커널과 실시간 특성들의 추출없이 그때그때 스케줄하는 동적(dynamic) 커널이 그들이다[31].

정적 커널의 특징은 시스템의 동작을 미리 예측하여 스케줄하므로써 시스템의 상태에 적합한 스케줄을 할 수 있으며, 정확한 스케줄을 통하여 자원을 보다 더 효율적으로 사용할 수 있다는 장점을 갖는다. 그러나 멀티미디어 응용의 요구를 미리 알 수 있어야 한다는 문제점을 가지며 규모가 작은, 요구들이 잘알려진, 특정한 응용을 위한 운영체제에 적합하다. 이러한 정적 커널의 예로서 Intel의 AVK[11]를 들 수 있다.

이에 반하여, 동적 커널은 시스템의 동작을 미리 알기 어려운 일반적인 시스템에서 취할 수 있는 접근방법으로서 미래에 대한 예측없이 가장 시급한 요구를 우선적으로 처리하는 고정우선순위 스케줄링 기법을 주로 사용하게 된다. 동적 커널은 시스템내의 자원을 효율적으로 사용하기 어렵고 동적으로 재스케줄하는 부담과 선점을 위한 부담을 진다는 문제점을 갖고 있으나 다양

표 1 커널의 특성과 관련 연구들

	동적 커널	정적 커널
예측성	요구에 대한 정보 적음	요구에 대한 상세한 정보
결합도	시스템의 부분간에 결합도 낮음	부분간의 결합도 높음
스케줄	고정우선 순위 스케줄링	여러가지의 실시간 스케줄링
장점	비주기적, 비예측적 사건 처리 가능	자원의 효율적인 이용
단점	재스케줄링, 선점을 위한 부담	비예측적 사건 처리 어려움
	비효율적인 자원 이용	예측성이 낮은 응용에 적용 곤란
구현	기존의 범용운영체제의 확장	새로운 운영체제 작성
시스템 예	Mach의 확장[1], UNIX의 확장[2] Real-Time Mach[4]	AVK[3]
관련연구	디스크 헤드 스케줄링[5, 6, 7, 8] 처리기 스케줄링[4] 통신채널 스케줄링[10, 11, 12]	디스크 헤드 스케줄링[5, 6, 7, 8] 처리기 스케줄링[9] 통신채널 스케줄링[13, 14]



한 요구를 처리할 수 있으므로 멀티미디어 컴퓨팅의 실시간 요구를 미리 분석하기 어려운 일반적인 시스템에 적합하다. 동적 커널의 예로는 기존의 운영체제를 멀티미디어 처리에 적합하도록 수정한 Mach의 수정[2], UNIX의 수정[4], Real-Time Mach[21] 등을 들 수 있다.

시스템의 각 단계의 관련 연구들도 이러한 관점에서 분류할 수 있다. 먼저 디스크 헤드 스케줄링 기법들[14,15,16,18]은 정적 커널과 동적 커널에 모두 적합한 스케줄링 기법이라고 할 수 있다. 처리기 스케줄링에서 Mercer[21]의 연구는 동적 커널의 대표적인 연구라고 할 수 있으며, Mei[22]의 연구는 정적 커널을 위한 스케줄링 기법이다. 통신 채널 스케줄링에서는 Zheng[24], Baek[26]의 연구 등이 정적 커널에 적합한 스케줄링이라고 할 수 있으며, 통신채널의 과부하시에 데이터양을 줄이는 흐름제어(flow) 기법들[27,28,29]은 동적 커널기법이라고 할 수 있다.

표 1에서는 정적, 동적 커널의 특징들과, 관련 연구들을 정리하여 나타내고 있다. 표에서 시스템의 각 부분간의 결합도란 화일시스템, 중앙처리기, 통신처리기 등의 각 부분들이 연속미디어들의 시간제약을 유지하기 위하여 얼마나 유기적으로 결합되어야 하는가를 나타낸다. 정적커널의 경우에는 각 부분간에 데이터를 전달하는 시점과 양이 분명히 정의되고 지켜지지만, 동적 커널에서는 잘 지켜지지 않는다.

## 5. 결론 및 향후연구방향

본 기고에서는 요즈음 각광을 받고 있는 멀티미디어 컴퓨팅을 위하여 운영체제에서 제공해야 하는 기능들을 정의하고 관련연구들을 조사하였다. 그 결과 멀티미디어 컴퓨팅에서 데이터양의 방대함, 연속미디어의 연속성, 멀티미디어 내의 시간적 동기화 등이 문제가 됨을 보였으며, 이들 문제를 풀기 위한 관련 연구들이 실시간 스케줄링과 같은 실시간 기술들을 이용하고 있음을 보였다. 현재까지 멀티미디어 컴퓨팅을 목표로 독자적으로 설계된 운영체제는 Intel의 AVK 정도로 적었으며, 대신에 기존의 운영체제를 바탕으로 멀티미디어 컴퓨팅을 위한 기능들을

확장한 것들이 많이 있었다.

향후연구방향으로는 대화형(Interactive) VOD 등의 응용들처럼 요구의 명세를 분명히 할 수 있는 경우에 AVK와 같은 특화된, 작은 운영체제의 수요가 더욱 많아질 것으로 예상되므로 이에 관한 연구가 지속되어야 할 것이다.

## 참고문헌

- [1] Sheng-Chang Cheng and John A. Stankovic. Scheduling algorithms for hard real-time systems - a brief survey. In Tutorial Hard-Real Time Systems, pp. 150~173. IEEE Computer Society Press, 1988.
- [2] J. Nakajima, M. Yazaki, and H. Matsumoto. Multimedia/realtime extensions for the mach operating system. In Proc. of the Summer 1991 Usenix Conference, pp. 183~198, Jun. 1991.
- [3] M. Pasiaka, P. Crumley, and A. Marks. Distributed multimedia: How can the necessary data rates be supported. In Proc. of the Summer 1991 Usenix Conference, pp. 169~182, Jun. 1991.
- [4] K. Fall and J. Pasquale. Improving continuous-media playback performance with in-kernel data paths. In Proc. of IEEE International Conference on Multimedia Computing and System, pp. 100~109, May 1994.
- [5] P. V. Rangan and H. M. Vin. Designing file systems for digital video and audio. In Proc. of the 13 ACM Symposium on Operating Systems Principles, pages 81~94, Oct. 1991.
- [6] R. Steinmetz. Synchronization properties in multimedia systems. IEEE Journal on Selected Areas in Communications, 8(3):401~412, Apr. 1990.
- [7] L. Lamont and N. D. Georganas. Synchronization architecture and protocols for a multimedia news service application. In Proc. of IEEE International Conference on Multimedia Computing and System, pp. 3~8. May 1994.
- [8] D. L. Gall. Mpeg: A video compression standard for multimedia applications. Communications of the ACM, 34(4):46~58, Apr. 1991.
- [9] M. J. Bach. The Design of the UNIX Operating System. Prentice-Hall, Inc., Englewood Cliffs,

- New Jersey, 1986.
- [10] H. M. Dietel. An Introduction to Operating Systems(2nd Ed.). Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.
- [11] J. L. Green. The evolution of dvi system software. Communications of the ACM, 35(1):36~51, Jul. 1992.
- [12] Intel Corp. ActionMedia 750 Reference Manual. 1991.
- [13] J. Wells, Q. Yang, and C. Yu. Placement of audio data on optical disks. In International Conference on Multimedia Information Systems, pp. 123~134, 1991.
- [14] A. L. N. Reddy and J. Wyllie. Disk scheduling in a multimedia i/o system. In Proc. of ACM Multimedia'93, pp. 225~234, Aug. 1993.
- [15] M. -S. Chen, D. D. Kandlur, and S. Y. Yu. Optimization of the grouped sweeping scheduling (gss) with heterogeneous multimedia streams. In Proc. of ACM Multimedia'93, pp. 235~242, Aug. 1993.
- [16] H. M. Vin, Al. Goyal, An. Goyal, and P. Goyal. An observation-based admission control algorithm for multimedia servers. In Proc. of IEEE International Conference on Multimedia Computing and System, pp. 234~244, May 1994.
- [17] M. Garey and D. Johnson. Computers and Intractability. W.H. Freeman and Company, 1979.
- [18] C. S. Lim and S. C. Kim. Efficient storage subsystem configuration for video on-demand server. In Proc. of IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications, pp. 22~26, Aug. 1994.
- [19] S. Chen and J. A. Stankovic. Performance evaluation of two new disk scheduling algorithms for real-time systems. Realtime Systems Journal, Sep. 1991.
- [20] Kenchamma-Hosekote D. R. and J. Srivastava. Scheduling continuous media in a video-on-demand server. In Proc. of IEEE International Conference on Multimedia Computing and System, pp. 19~29, May 1994.
- [21] C.W. Mercer, S. Savage, and H. Tokuda. Process capacity reserves: Operating system support for multimedia applications. In Proc. of IEEE International Conference on Multimedia Computing and System, pp. 90~99, May 1994.
- [22] H. Mei. Scheduling multimedia tasks with timing constraints on distributed systems. In Proc. of IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications, pp. 180~185, Aug. 1994.
- [23] Y. S. Ryu and K. Koh. Considering run-time overheads in the scheduling for continuous-media task systems. In Proc. of IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications, pp. 186~189, Aug. 1994.
- [24] Zheng. Q. Real-time Fault-tolerant Communication in Computer Networks. Phd thesis, University of Michigan, 1993.
- [25] K. Jeffay, D. F. Stanat, and C. U. Martel. On non-preemptive scheduling of periodic and sporadic tasks. In Proceedings of Real-Time Systems Symposium, pp. 129~139, 1991.
- [26] Y. Baek and K. Koh. Schedulability analysis on non-preemptive real-time tasks for continuous media retrieval. Journal of the Korea Information Science Society, 21(7):1252~1260, Jul. 1994.
- [27] A. Park and P. English. A variable rate strategy for retrieving audio data from secondary storage. In International Conference on Multimedia Information Systems, pp. 135~146, 1991.
- [28] D. P. Anderson, R. Govindan, and G. Homsy. Abstractions for continuous media in a network window system. In International Conference on Multimedia Information Systems, pp. 273~298, 1991.
- [29] S. Chakrabarti and R. Wang. Adaptive control for packet video. In Proc. of IEEE International Conference on Multimedia Computing and System, pp. 56~62, May 1994.
- [30] C. Kim and S. W. Kang. A media synchronization scheme for distributed multimedia systems. In Proc. of IASTED/ISMM International Conference on Distributed Multimedia Systems and Applications, pp. 163~166, Aug. 1994.
- [31] P. G. Jansen and P. Sibben. Real-time in multimedia: Opportunistic scheduling or quality of service contracts? In Proc. of IASTED/ISMM

International Conference on Distributed Multimedia Systems and Applications, pp. 194~200, Aug. 1994.

**김진천**



1983 한양대학교 전기공학과 학사  
 1985 미시간 주립대학교 전자공학과 석사  
 1990 ~현재 한국과학기술원 전산학과 박사과정 재학 중  
 1988 ~현재 삼성전자 연구원  
 관심 분야: 병렬처리, 고성능 네트워크, 멀티미디어 컴퓨팅, 컴퓨터 구조.

**윤현수**



1979 서울대학교 공과대학 전자공학과 학사  
 1981 한국과학기술원 전산학과 석사  
 1981 ~1984 삼성전자 연구원  
 1988 오하이오 주립대학 전산학 박사  
 1988 ~1989 AT&T Bell Labs. 연구원  
 1989 ~현재 한국과학기술원 전산학과 교수

관심 분야: 병렬 컴퓨터 구조, ATM망 기술, 멀티미디어 컴퓨팅.

**은성배**



1979 서울대학교 전자계산기공학과 학사  
 1987 한국과학기술원 전산학과 석사  
 1990 ~현재 한국과학기술원 전산학과 박사과정 재학 중  
 1987 ~1990 전자통신연구소 연구원  
 관심 분야: 멀티미디어 컴퓨팅, 실시간 시스템.

● 논문모집 ●

제 7회 영상처리 및 이해에 관한 워크숍

- 대회일자 : 1995년 2월 3일~4일
- 대회장소 : 서강대학교 다산관
- 논문마감 : 1994년 11월 30일
- 제출·문의 : 송실대 최형일 교수

Tel : (02) 820-0679

Fax : (02) 814-3627