



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2011년09월27일  
(11) 등록번호 10-1067397  
(24) 등록일자 2011년09월19일

(51) Int. Cl.  
G06F 3/16 (2006.01)  
(21) 출원번호 10-2004-0018840  
(22) 출원일자 2004년03월19일  
심사청구일자 2009년02월25일  
(65) 공개번호 10-2004-0082993  
(43) 공개일자 2004년09월30일  
(30) 우선권주장  
10/393,661 2003년03월20일 미국(US)  
(56) 선행기술조사문헌  
US6490722 B1

(73) 특허권자  
마이크로소프트 코포레이션  
미국 워싱턴주 (우편번호 : 98052) 레드몬드 원  
마이크로소프트 웨이  
(72) 발명자  
스웬손,스티븐이.  
미국98052워싱턴주레드몬드엔이73번에스티15338  
플렌니켄, 데이비드더블유.  
미국98053워싱턴주레드몬드엔이34번에스티.26225  
(74) 대리인  
제일특허법인

전체 청구항 수 : 총 34 항

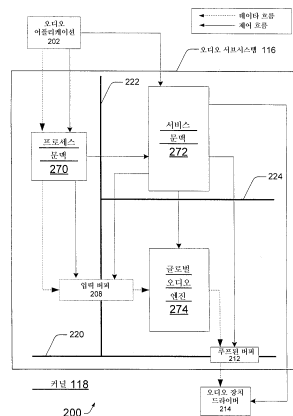
심사관 : 한선경

(54) 오디오 프로세싱 오브젝트에 의한 액세스 위반에 대한 복구를 수행하기 위한 컴퓨팅 시스템 동작 방법, 컴퓨터-판독가능 매체 및 컴퓨팅 시스템

(57) 요약

오퍼레이팅 시스템은 커널을 위한 파티션과, 논리적으로 구분된 서브파티션들을 갖는 오디오 서브시스템을 위한 파티션으로 논리적으로 구분되어 있다. 오디오 어플리케이션은 오디오 서브시스템을 실행함으로써 제1 오디오 데이터 스트림을 생성한다. 제1 오디오 데이터 스트림을 프로세싱하기 위한 프로세스의 문맥은 제1 서브파티션에서 보존된다. 그 프로세스에 의한 제1 오디오 데이터 스트림의 프로세싱은 제2 서브파티션에서 일어난다. 프로세스가 액세스 위반을 일으키면, 프로세스는 오디오 서브시스템에서 제1 서브파티션으로부터 보존된 문맥을 복구함으로써 회복된다.

대표도 - 도2



**특허청구의 범위**

**청구항 1**

오디오 데이터 스트림을 생성하는 오디오 어플리케이션(202, 528, 558)을 실행하고 오퍼레이팅 시스템(114, 526)을 실행시키기 위한 프로세서(504)를 포함하는 컴퓨팅 시스템(100, 200, 500)을 동작시키는 방법에 있어서,

상기 오퍼레이팅 시스템은 커널(kernel)(118)을 위한 파티션(partition)과, 논리적으로 구분된 서브파티션들(subpartition)(270, 272, 274)을 갖는 오디오 서브시스템(116)을 위한 파티션으로 논리적으로 구분되고,

상기 방법은,

제1 오디오 데이터 스트림을 상기 오디오 서브시스템으로 처리하기 위한 프로세스의 문맥을 제1 서브파티션 내에 보존한 다음에, 제2의 상기 서브파티션 내에서 상기 프로세스를 실행하는(308, 310) 단계; 및

상기 프로세스가 액세스 위반(access violation)을 일으키면(312), 상기 제1 서브파티션으로부터 상기 문맥을 복구(restore)함으로써 상기 프로세스를 회복(recover)하는 단계

를 포함하는 것을 특징으로 하는 컴퓨팅 시스템 동작 방법.

**청구항 2**

제1항에 있어서,

상기 오퍼레이팅 시스템에게 오디오 장치 드라이버에 의해 구동되는 오디오 장치에서 출력될 상기 오디오 데이터 스트림에 대해 수행될 글로벌 오디오 이펙트(global audio effect)를 위한 오디오 프로세싱 오브젝트(audio processing object), 즉, APO를 포함하여 프로세스를 실행하도록 하는 요청을 수신하는 단계 - 상기 글로벌 오디오 이펙트는 상기 오디오 장치에 전달(direct)되는 모든 오디오에 영향을 주고, 상기 오디오 서브시스템은 입력 버퍼(208), 루프된 버퍼(looped buffer), 및 서비스 문맥(272) 및 글로벌 오디오 엔진(274)을 포함하는 논리적으로 구분된 상기 서브파티션들을 포함하고, 상기 글로벌 오디오 엔진은 상기 글로벌 오디오 엔진이 상기 루프된 버퍼를 통해 통신하는 상기 오디오 장치 드라이버에 의해 구동되는 오디오 장치를 위한 프록시임 -

상기 서비스 문맥 내에 상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼에 대한 문맥을 보존하는 단계(302, 306, 404);

상기 루프된 버퍼 내에 디지털 신호 프로세싱, 즉, DSP 오디오 데이터 스트림을 형성하기 위해 상기 입력 버퍼 내의 오디오 데이터 스트림에 대해 상기 APO를 이용하여 디지털 신호 프로세싱을 수행하도록(310) 상기 글로벌 오디오 엔진을 동작시키는 단계; 및

상기 루프된 버퍼 내의 상기 DSP 오디오 데이터 스트림을 상기 오디오 장치를 구동하는 상기 오디오 장치 드라이버에 입력하는 단계

를 더 포함하고,

상기 회복하는 단계는,

상기 APO의 종료(312) 또는 상기 APO에 의한 액세스 위반(312)시에,

상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼 각각을, 상기 서비스 문맥 내에 각각에 대하여 각기 보존된 문맥을 사용하여 복구하는 단계(314); 및

상기 글로벌 오디오 엔진에서 상기 APO의 실행을 선택적으로 다시 시작하는 단계(308)

를 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 3**

제2항에 있어서,

상기 APO의 종료 또는 상기 APO에 의한 액세스 위반은 상기 서비스 문맥에 의해 검출(412)되는 컴퓨팅 시스템

동작 방법.

**청구항 4**

제2항 또는 제3항에 있어서,

상기 입력 버퍼의 문맥은 상기 입력 버퍼와 관련된 주소 및 데이터 구조를 포함하고,

상기 글로벌 오디오 엔진의 문맥은 상기 글로벌 오디오 엔진과 관련된 주소 및 데이터 구조를 포함하고,

상기 루프된 버퍼의 문맥은 상기 루프된 버퍼와 관련된 주소 및 데이터 구조를 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 5**

제2항 또는 제3항에 있어서,

상기 APO에 의한 액세스 위반시에,

상기 APO에 의한 액세스 위반의 수를 나타내는 카운터를 증분시키는 단계(318); 및

상기 카운터가 소정의 한계를 초과할 때, 진단 메시지(diagnostic)를 출력(322a)하고, 상기 글로벌 오디오 엔진에서 상기 APO의 실행을 다시 시작하는 것을 금지(322b)하는 단계

를 더 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 6**

제2항 또는 제3항에 있어서,

상기 오퍼레이팅 시스템은 복수의 상기 글로벌 오디오 엔진에 대응하는 복수의 상기 루프된 버퍼를 더 포함하고,

상기 방법은,

각각의 상기 글로벌 오디오 엔진에 의해, 대응하는 상기 오디오 장치를 구동하는 대응하는 상기 오디오 장치 드라이버로의 입력을 제공하기 위하여, 대응하는 상기 루프된 버퍼에 하나의 상기 DSP 오디오 데이터 스트림을 출력(310)하는 단계

를 더 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 7**

제6항에 있어서,

상기 오퍼레이팅 시스템은 대응하는 상기 오디오 장치에 의해 렌더링될 오디오 데이터 스트림을 각각 포함하는 복수의 상기 입력 버퍼를 더 포함하고,

각각의 상기 글로벌 오디오 엔진은 하나의 상기 입력 버퍼에 대응하는 컴퓨팅 시스템 동작 방법.

**청구항 8**

제7항에 있어서,

상기 입력 버퍼 각각에 의해, 상기 오퍼레이팅 시스템으로부터 적어도 하나의 상기 오디오 장치에 대한 오디오 서비스를 요청하는 오디오 어플리케이션으로부터 대응하는 상기 오디오 데이터 스트림을 수신하는 단계를 더 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 9**

제2항 또는 제3항에 있어서,

상기 오디오 장치 드라이버에 의해 구동되는 상기 오디오 장치에 의해 상기 DSP 오디오 데이터 스트림을 렌더링하는 단계를 더 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 10**

제2항 또는 제3항에 있어서,

상기 오퍼레이팅 시스템은 상기 커널을 위한 파티션, 및 상기 논리적으로 구분된 서브파티션들을 갖는 상기 오디오 서브시스템을 위한 파티션을 포함하는 복수의 구분된 파티션을 갖는 메모리(516, 506) 내에 있고,

상기 제1 서브파티션은 상기 서비스 문맥을 위한 것이고,

상기 제2 서브파티션은 상기 글로벌 오디오 엔진을 위한 것인 컴퓨팅 시스템 동작 방법.

**청구항 11**

제10항에 있어서,

상기 메모리 내의 오디오 서브시스템의 파티션은 상기 입력 버퍼에 상기 오디오 데이터 스트림을 출력하기 위해 오디오 프로세싱 오브젝트, 즉, APO를 실행(406)하는 로컬 이펙트 엔진을 갖는 프로세싱 문맥(270)을 위한 논리적으로 구분된 서브 파티션을 더 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 12**

제11항에 있어서,

상기 입력 버퍼 내의 상기 오디오 데이터 스트림은 상기 오디오 장치로 각각 렌더링될 복수의 오디오 어플리케이션(202, 528, 558)으로부터의 오디오 데이터를 포함하고,

상기 방법은,

상기 루프된 버퍼 내에 상기 DSP 오디오 데이터 스트림을 형성하기 위해, 상기 글로벌 오디오 엔진에 의해, 상기 복수의 오디오 어플리케이션으로부터의 상기 오디오 데이터를 혼합(mixing)(408)하는 단계

를 더 포함하는 것을 특징으로 하는 컴퓨팅 시스템 동작 방법.

**청구항 13**

제2항 또는 제3항에 있어서,

상기 글로벌 오디오 이펙트는 잔향 효과(reverberation effect), 오디오 왜곡 효과(audio distortion effect), 스피커 보정 효과(speaker compensation effect), 멀티밴드 오디오 이퀄라이저 효과(multiband audio equalizer effect)로 이루어진 그룹으로부터 선택되는 컴퓨팅 시스템 동작 방법

**청구항 14**

제2항 또는 제3항에 있어서,

상기 글로벌 오디오 엔진은 상기 APO를 실행시키기 위한 것이고,

상기 수신하는 단계는 오디오 어플리케이션으로부터 상기 요청을 수신하는 단계를 포함하고,

상기 동작시키는 단계는 상기 루프된 버퍼에 상기 DSP 오디오 데이터 스트림을 출력(310)하도록 상기 글로벌 오디오 엔진을 동작시키는 단계를 포함하고,

상기 입력하는 단계는 상기 DSP 오디오 데이터스트림을 렌더링하기 위해 상기 오디오 장치를 구동시키는 상기 오디오 장치 드라이버에 상기 루프된 버퍼 내의 DSP 오디오 데이터 스트림을 입력하는 단계를 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 15**

제10항에 있어서,

상기 입력 버퍼는 상기 프로세스 문맥의 파티션과 상기 글로벌 오디오 엔진의 파티션 간에 공유된 상기 메모리 내의 상기 오디오 서브시스템의 파티션에 논리적으로 설치되고,

상기 루프된 버퍼는 상기 글로벌 오디오 엔진의 파티션과 상기 커널의 파티션 간에 공유된 상기 메모리 내의 상

기 오디오 서브시스템의 파티션에 논리적으로 설치되는 컴퓨팅 시스템 동작 방법.

**청구항 16**

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 보존된 프로세스의 문맥은, 상기 오디오 서브시스템 내의 상기 보존된 문맥의 복구에 의해 상기 프로세스를 회복하는 데 충분한 주소 및 데이터 구조 정보를 포함하는 컴퓨팅 시스템 동작 방법.

**청구항 17**

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 프로세서가 상기 오디오 어플리케이션을 실행하여, 상기 제1 오디오 데이터 스트림이 출력될 제1 오디오 장치와는 다른 제2 오디오 장치로 출력될 제2 오디오 데이터 스트림을 생성할 때, 상기 프로세스의 액세스 위반 및 회복은,

상기 제2 오디오 어플리케이션의 실행, 및 상기 제2 오디오 장치로 출력될 상기 제2 오디오 데이터 스트림에 영향을 주지 않는 컴퓨팅 시스템 동작 방법.

**청구항 18**

컴퓨팅 시스템(100, 200, 500)에 의해 실행될 때, 오퍼레이팅 시스템(114, 526)을 실행시킴으로써 제1 오디오 데이터 스트림을 생성하라는 오디오 어플리케이션(202, 528, 558)으로부터의 요청을 처리하는 단계를 포함하는 방법을 구현하는 컴퓨터-판독가능 명령어들을 갖는 하나 이상의 컴퓨터-판독가능 기록 매체(506, 516, 520, 524)로서,

상기 오퍼레이팅 시스템은 커널(118)을 위한 파티션과, 논리적으로 구분된 서브파티션들(270, 272, 274)을 갖는 오디오 서브시스템(116)을 위한 파티션으로 논리적으로 구분되고,

상기 방법은,

제1 오디오 데이터 스트림을 상기 오디오 서브시스템으로 처리하기 위한 프로세스의 문맥을 제1 서브파티션 내에 보존한 다음에, 제2의 상기 서브파티션 내에서 상기 프로세스를 실행하는 단계; 및

상기 프로세스가 액세스 위반을 일으키면(312), 상기 제1 서브파티션으로부터 상기 문맥을 복구함으로써 상기 프로세스를 회복하는 단계

를 더 포함하는 것을 특징으로 하는 하나 이상의 컴퓨터-판독가능 기록 매체.

**청구항 19**

제18항에 있어서,

상기 보존된 프로세스의 문맥은, 상기 오디오 서브시스템 내의 상기 보존된 문맥의 복구에 의해 상기 프로세스를 회복하는 데 충분한 주소 및 데이터 구조 정보를 포함하는 하나 이상의 컴퓨터-판독가능 기록 매체.

**청구항 20**

제18항 또는 제19항에 있어서,

오퍼레이팅 시스템을 실행시킴으로써 제2 오디오 데이터 스트림을 생성하라는 오디오 어플리케이션(202, 528, 558)으로부터의 요청을 처리하기 위한 컴퓨터-판독가능 명령어들을 더 포함하고,

상기 제2 오디오 데이터 스트림은 상기 제1 오디오 데이터 스트림이 출력될 제1 오디오 장치와는 다른 제2 오디오 장치로 출력되도록 지향되고,

상기 프로세스의 액세스 위반 및 회복은, 상기 제2 오디오 어플리케이션의 실행에 의한 상기 제2 오디오 데이터 스트림의 생성, 및 상기 제2 오디오 장치로 출력될 상기 제2 오디오 데이터 스트림에 영향을 주지 않는 하나 이상의 컴퓨터-판독가능 기록 매체.

**청구항 21**

제18항 또는 제19항에 있어서,

상기 제1 서브파티션은 서비스 문맥(272)을 포함하고, 상기 제2 서브파티션은 글로벌 오디오 엔진(274)을 포함하고,

상기 오디오 서브시스템은 입력 버퍼(208), 및 오디오 장치를 구동하기 위한 장치 드라이버(214)와 통신하는 루프된 버퍼(212)를 더 가지고,

상기 글로벌 오디오 엔진은 상기 글로벌 오디오 엔진이 상기 루프된 버퍼를 통해 통신하는 상기 오디오 장치 드라이버에 의해 구동되는 상기 오디오 장치를 위한 프록시이고,

상기 방법은,

상기 장치 드라이버에 의해 구동되는 상기 오디오 장치에 의해 렌더링될 상기 루프된 버퍼 내에 상기 제1 오디오 데이터 스트림을 형성하기 위해, 상기 글로벌 오디오 엔진에 의해, 상기 입력 버퍼 내의 오디오 데이터 스트림에 대해 수행될 글로벌 오디오 이펙트를 위한 오디오 프로세싱 오브젝트, 즉, APO를 실행시키는(304, 308, 310) 단계;

상기 서비스 문맥 내에, 상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼의 문맥을 보존(302, 306, 404)하는 단계;

상기 글로벌 오디오 엔진에서 실행되는 상기 APO의 종료(312) 또는 상기 APO에 의한 액세스 위반(312)시에,

상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼 각각을, 상기 서비스 문맥 내에 각각에 대하여 각기 보존된 문맥을 사용하여 복구하는 단계(314); 및

상기 글로벌 오디오 엔진에서 상기 APO의 실행을 선택적으로 다시 시작하는 단계(308)

를 더 포함하는 하나 이상의 컴퓨터-판독가능 기록 매체.

#### 청구항 22

제21항에 있어서,

상기 입력 버퍼는 각각 상기 오디오 장치로 렌더링될 복수의 오디오 어플리케이션(202, 528, 558)으로부터의 오디오 데이터를 저장하고,

상기 방법은,

상기 루프된 버퍼 내에 오디오 데이터를 형성하기 위해, 상기 글로벌 오디오 엔진에 의해 상기 복수의 오디오 어플리케이션으로부터의 상기 오디오 데이터를 혼합하는 단계(408)

를 더 포함하는 하나 이상의 컴퓨터-판독가능 기록 매체.

#### 청구항 23

제18항 또는 제19항에 있어서,

제2항 또는 제3항의 방법을 수행하기 위한 컴퓨터-판독가능 명령어들을 더 갖는 하나 이상의 컴퓨터-판독가능 기록 매체.

#### 청구항 24

제1 오디오 데이터 스트림을 생성하도록 오디오 어플리케이션(202, 528, 558)을 실행하고 오퍼레이팅 시스템(114, 526)을 실행시키기 위한 프로세서(504)를 포함하는 컴퓨팅 시스템(100, 200, 500)에 있어서,

상기 오퍼레이팅 시스템은 커널(118)을 위한 파티션과, 논리적으로 구분된 서브파티션들(270, 272, 274)을 갖는 오디오 서브시스템(116)을 위한 파티션으로 논리적으로 구분되고,

상기 컴퓨팅 시스템은

제1 오디오 데이터 스트림을 상기 오디오 서브시스템으로 처리하기 위한 프로세스의 문맥을 제1 서브파티션 내에 보존(306)한 다음에, 제2의 상기 서브파티션 내에서 상기 프로세스를 실행(308, 310)시키기 위한 수단; 및

상기 프로세스가 액세스 위반(312)을 일으키면, 상기 제1 서브파티션으로부터 상기 문맥을 복구함으로써 상기 프로세스를 회복하기 위한 수단

을 더 포함하는 것을 특징으로 하는 컴퓨팅 시스템.

**청구항 25**

제24항에 있어서,

상기 보존된 프로세스의 문맥은, 상기 오디오 서브시스템 내의 상기 보존된 문맥의 복구에 의해 상기 프로세스를 회복하는 데 충분한 주소 및 데이터 구조 정보를 포함하는 컴퓨팅 시스템.

**청구항 26**

제24항 또는 제25항에 있어서,

상기 프로세서는 또한 제2 오디오 어플리케이션(202, 528, 558)을 실행하여, 상기 제1 오디오 데이터 스트림이 출력될 제1 오디오 장치와는 다른 제2 오디오 장치로 출력될 제2 오디오 데이터 스트림을 생성하기 위한 것이고,

상기 프로세스의 액세스 위반 및 회복은, 상기 제2 오디오 어플리케이션의 실행, 및 상기 제2 오디오 장치로 출력될 상기 제2 오디오 데이터 스트림에 영향을 주지 않는 컴퓨팅 시스템.

**청구항 27**

제24항 또는 제25항에 있어서,

상기 논리적으로 구분된 파티션들을 저장(506, 516)하기 위한 수단

- 상기 오디오 서브시스템은,

입력 버퍼(208), 및 오디오를 렌더링하기 위한 수단을 구동하기 위한 수단과 통신하는 루프된 버퍼(212); 및

상기 논리적으로 구분된 서브파티션들

-상기 제1 서브파티션은 서비스 문맥(272)을 포함하고, 상기 제2 서브파티션은 글로벌 오디오 엔진(274)을 포함하고, 상기 글로벌 오디오 엔진은 상기 글로벌 오디오 엔진이 상기 루프된 버퍼를 통해 통신하는 상기 구동 수단에 의해 구동되는 오디오를 렌더링하기 위한 상기 수단을 위한 프록시임 -

을 포함함 -

상기 오디오 서브시스템을 실행할 때,

상기 서비스 문맥 내에 상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼의 문맥을 보존하고(302, 306, 404),

상기 오디오 장치를 구동하기 위한 상기 수단에 의해 구동되는 오디오를 렌더링하기 위한 수단에 의해 렌더링될 오디오 데이터 스트림을 상기 루프된 버퍼 내에 형성하기 위해 상기 입력 버퍼 내의 오디오 데이터 스트림에 대해 글로벌 오디오 이펙트를 생성하도록, 오디오 프로세싱 오브젝트, 즉, APO를 실행(304, 308, 310)시키기 위한 수단; 및

상기 글로벌 오디오 엔진 내에서 실행되는 상기 APO에 의한 액세스 위반(312) 시에,

상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼 각각을, 상기 서비스 문맥 내에 각각에 대하여 각기 보존된 문맥을 사용하여 복구하고(314),

상기 글로벌 오디오 엔진에서 상기 APO의 실행을 다시 시작(308)하기 위한 수단

을 더 포함하는 컴퓨팅 시스템.

**청구항 28**

제27항에 있어서,

상기 오디오 서브시스템은 대응하는 복수의 상기 글로벌 오디오 엔진에 대하여 복수의 상기 루프된 버퍼를 더

포함하고,

각각의 상기 글로벌 오디오 엔진은 오디오를 렌더링하기 위한 대응하는 상기 수단을 구동하기 위한 대응하는 상기 수단으로의 입력을 제공하기 위하여, 대응하는 상기 루프된 버퍼에 하나의 상기 오디오 데이터 스트림을 출력하는 컴퓨팅 시스템.

**청구항 29**

제28항에 있어서,

상기 오디오 서브시스템은 오디오를 렌더링하기 위한 대응하는 상기 수단에 의해 렌더링될 오디오 데이터 스트림을 각각 포함하는 복수의 상기 입력 버퍼를 더 포함하고,

각각의 상기 글로벌 오디오 엔진은 하나의 상기 입력 버퍼에 대응하는 컴퓨팅 시스템.

**청구항 30**

제29항에 있어서,

각각의 상기 입력 버퍼는, 상기 오퍼레이팅 시스템으로부터 오디오를 렌더링하는 적어도 하나의 상기 수단에 대한 오디오 서비스를 요청하는 오디오 어플리케이션으로부터 대응하는 상기 오디오 데이터 스트림을 수신하는 컴퓨팅 시스템.

**청구항 31**

제27항에 있어서,

상기 오디오 서브시스템의 상기 논리적으로 구분된 복수의 서브파티션은, 상기 오디오 데이터 스트림을 상기 입력 버퍼로 출력하기 위해 오디오 프로세싱 오브젝트, 즉, APO를 실행하는 로컬 이펙트 엔진을 갖는 프로세싱 문맥(270)을 위한 서브파티션을 더 포함하는 컴퓨팅 시스템.

**청구항 32**

제31항에 있어서,

상기 입력 버퍼 내의 오디오 데이터 스트림은 오디오를 렌더링기 위한 상기 수단에 의해 각각 렌더링될 복수의 오디오 어플리케이션(202, 528, 558)으로부터의 오디오 데이터를 포함하고,

상기 글로벌 오디오 엔진은 상기 루프된 버퍼 내에 상기 오디오 데이터 스트림을 형성하기 위해 상기 복수의 오디오 어플리케이션으로부터의 상기 오디오 데이터를 혼합(408)하는 컴퓨팅 시스템.

**청구항 33**

제27항에 있어서,

상기 프로세서는 또한, 복수의 오디오 어플리케이션(202, 528, 568)을 실행하기 위한 것이고,

상기 컴퓨팅 시스템은,

상기 오퍼레이팅 시스템에 전달하기 위한 오디오 데이터를 각각 생성하기 위해 상기 복수의 오디오 어플리케이션과 상호 동작(interoperating)하는 복수의 멀티미디어 관독 장치;

상기 오퍼레이팅 시스템에 전달되는 상기 오디오 데이터를 렌더링하기 위해 각 오디오 장치 드라이버(214)에 의해 구동되는 복수의 오디오 장치; 및

상기 복수의 오디오 어플리케이션 및 상기 오퍼레이팅 시스템을 위한 저장 장치를 포함하고,

상기 오디오 서브시스템은 프로세스 문맥(270)을 위한 논리적으로 구분된 서브파티션을 더 갖고,

상기 입력 버퍼는 상기 프로세스 문맥 및 상기 글로벌 오디오 엔진과 통신하고,

상기 오디오 서브시스템은 상기 오디오 장치들 중 하나를 구동하기 위한 상기 오디오 장치 드라이버들 중 하나와 통신하는 루프된 버퍼(212)를 더 갖고,

상기 프로세서는,

상기 서비스 문맥 내에, 상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼를 위한 문맥을 보존(314)하고,

하나의 상기 오디오 어플리케이션으로부터, 상기 입력 버퍼의 오디오 데이터 스트림으로서의 출력에 대한 상기 프로세스 문맥으로의 입력을 위한 오디오 데이터를 생성하고,

상기 오디오 장치 드라이버들 중 하나에 의해 구동되는 상기 오디오 장치들 중 하나에 의해 렌더링될 상기 오디오 데이터 스트림을 상기 루프된 버퍼 내에 형성하기 위하여, 상기 APO를 실행(304, 308, 310)하고,

상기 글로벌 오디오 엔진에서 실행되는 상기 APO에 의한 액세스 위반(312)시, 상기 복구하는 단계 및 다시 시작하는 단계를 수행하도록 상기 오퍼레이팅 시스템 및 상기 오디오 어플리케이션들 중 하나를 실행하는 컴퓨팅 시스템.

**청구항 34**

제24항 또는 제25항에 있어서,

상기 오퍼레이팅 시스템에게 오디오 장치 드라이버에 의해 구동되는 오디오 장치에서 출력될 상기 오디오 데이터 스트림에 대해 수행될 글로벌 오디오 이펙트를 위한 오디오 프로세싱 오브젝트, 즉, APO를 포함하여 프로세스를 실행하도록 하는 요청을 수신하기 위한 수단 - 상기 글로벌 오디오 이펙트는 상기 오디오 장치에 전달되는 모든 오디오에 영향을 주고, 상기 오디오 서브시스템은 입력 버퍼, 루프된 버퍼, 및 서비스 문맥 및 글로벌 오디오 엔진을 포함하는 논리적으로 구분된 상기 서브파트یشن들을 포함하고, 상기 글로벌 오디오 엔진은 상기 글로벌 오디오 엔진이 상기 루프된 버퍼를 통해 통신하는 상기 오디오 장치 드라이버에 의해 구동되는 오디오 장치를 위한 프록시임 -

상기 서비스 문맥 내에 상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼에 대한 문맥을 보존하기 위한 수단;

상기 루프된 버퍼 내에 디지털 신호 프로세싱, 즉, DSP 오디오 데이터 스트림을 형성하기 위해 상기 입력 버퍼 내의 오디오 데이터 스트림에 대해 상기 APO를 이용하여 디지털 신호 프로세싱을 수행하도록 상기 글로벌 오디오 엔진을 동작시키기 위한 수단; 및

상기 루프된 버퍼 내의 상기 DSP 오디오 데이터 스트림을 상기 오디오 장치를 구동하는 상기 오디오 장치 드라이버에 입력하기 위한 수단

을 더 포함하고,

상기 회복하기 위한 수단은,

상기 APO의 종료 또는 상기 APO에 의한 액세스 위반 시에,

상기 입력 버퍼, 상기 글로벌 오디오 엔진 및 상기 루프된 버퍼 각각을, 상기 서비스 문맥 내에 각각에 대하여 각기 보존된 문맥을 사용하여 복구하기 위한 수단; 및

상기 글로벌 오디오 엔진에서 상기 APO의 실행을 선택적으로 다시 시작하기 위한 수단

을 포함하는 컴퓨팅 시스템.

**청구항 35**

삭제

**청구항 36**

삭제

**청구항 37**

삭제

청구항 38

삭제

청구항 39

삭제

청구항 40

삭제

청구항 41

삭제

청구항 42

삭제

청구항 43

삭제

청구항 44

삭제

청구항 45

삭제

청구항 46

삭제

**명세서**

**발명의 상세한 설명**

**발명의 목적**

**발명이 속하는 기술 및 그 분야의 종래기술**

- [0016] 본 발명은 일반적으로 스트리밍 오디오 프로세싱과 관한 것으로, 더 상세하게는, 스트리밍 오디오 프로세싱 동안에 오디오 프로세싱 오브젝트에 의한 액세스 위반(access violation)으로부터의 복구에 관한 것이다.
- [0017] 퍼스널 컴퓨터(PC)와 같은 컴퓨팅 시스템이 점차 스트리밍 오디오 데이터를 기록하고 재생하는 데 사용되고 있다. PC는, 여러개의 오디오, 비디오, 및 멀티미디어 어플리케이션이 동시에 동작하는, 그 각각이 PC의 오퍼레이팅 시스템(OS)을 사용하는 미디어 서버의 한 유형으로서 사용될 수 있다. 미디어 서버로서의 PC 예로, 다음의 동시적인 사용 시나리오를 생각해보자. 디지털 비디오 디스크는 PC 상의 DVD 플레이어를 사용하는 제1 미디어 플레이어 어플리케이션(MP)에 의해 재생되고 있다. 전화 사용자가 다른 전화 사용자와 회선 교환 네트워크 상에서 이야기하는 것처럼 각 컴퓨팅 시스템의 사용자가 패킷 교환 네트워크 상에서 서로 이야기하는 네트워크 전화를 위한 리얼 타임 커뮤니케이션 어플리케이션(RTC)의 보이스 오버-인터넷-프로토콜(Voice-Over-Internet Protocol)을 위해, 마이크로폰이 사용되고 있다. 시스템 사운드는 사용자들이 PC와 상호 작용하는 동안 OS에 의해 생성되고 있다. 주문형 비디오 어플리케이션(VOD)은 스트리밍 멀티미디어가 케이블이나 위성 TV 방송 시스템을 통해 수신될 때 PC에 의해 실행된다. 음악 디스크(예를 들어, CD)는 PC의 CD-ROM 드라이브를 사용하는 제2 미디어 플레이어 어플리케이션의 실행에 의해 재생되고 있다. 다른 사용자가 PC 상에서 비디오 게임을 하고 있으며, 대응하는 게임 어플리케이션은 사용자와 게임 어플리케이션과의 상호 작용에 기초하여 오디오 데이터 스트림을 발생시키고 있다. 하나 이상의 글로벌 오디오 이펙트 어플리케이션(GFX)도 PC에서 실행될

수 있다. GFX의 예는 잔향 이펙트 어플리케이션, 오디오 왜곡 이펙트 어플리케이션, 하나 이상의 스피커에서 렌더링(rendering)된 스트리밍 오디오 데이터의 디지털 신호 프로세싱(DSP)에 의해 수행되는 스피커 보정 어플리케이션, 오디오 장치(예를 들면, 사운드 카드)와 함께 작용하는 사용자 인터페이스(UI)를 제공하여, PC 사용자가 다양한 프리셋(preset) (예를 들면, 콘서트 홀, 락, 클래식 등)을 위해 오디오 장치로부터의 오디오 출력을 조정할 수 있게 하도록 실행하는 멀티밴드 오디오 이퀄라이저 어플리케이션 및 하나 이상의 오디오 드라이버에 의해 구동되는 하나 이상의 오디오 장치로 출력되는 하나 이상의 오디오 데이터 스트림 상에 DSP를 행하는 다른 GFX를 포함한다.

[0018] 다른 오디오 비주얼 어플리케이션들 뿐만아니라 상기 예에서 나열된 임의의 어플리케이션은 불완전하게 기술되어 그 실행이 실패하고 어플리케이션이 중지될 수 있다. 어플리케이션의 그러한 실패는 PC의 OS에 PC 자체가 동작하는 것을 멈추는 컴퓨터 충돌로도 알려진 심각한 파손을 야기할 수 있다. 컴퓨터 충돌은 어플리케이션이 기술된 코드의 매우 심각한 소프트웨어 버그를 의미한다. OS에게 오디오 장치로 출력하라고 요구하는 오디오 어플리케이션의 예를 들면, 버그가 액세스 위반을 야기시킬 수 있다. OS의 커널(kernel)(예를 들면, 스케줄링 서비스)을 위한 메모리와 OS에서 오디오를 처리하는 데 사용되는 메모리는 서로 완전히 독립적인 것으로 가정된다. 액세스 위반은 오디오 어플리케이션이 상주하지 않는 메모리(예를 들면, 메모리 액세스가 불법이거나 허용되지 않는다)로 접근할 때 발생한다. 액세스 위반은 일반적으로 오디오 어플리케이션이 커널 모드에서 실행하는 오디오 장치 드라이버를 갖고 있을 때 일어난다. 만약 오디오 어플리케이션이 OS의 커널 공간으로 또는 오디오 장치 드라이버로의 액세스 위반을 일으키면, 액세스 위반은 컴퓨터 충돌을 야기할 수 있다.

[0019] OS의 커널이 손상되지 않는 한, 충돌되는 오디오 어플리케이션은 커널을 불안정하게 하지 않는다. 오히려, 오디오 어플리케이션만이 동작하는 것을 멈추고 다시 시작해야 한다. 그러나, 커널의 액세스 위반은 커널을 불안정하게 해서 복구가 불가능한 치명적인 에러를 발생시킬 수 있다. 마이크로소프트 윈도우? 오퍼레이팅 시스템 환경에서, PC 사용자는 파란 화면에 디스플레이되는 진단 메시지에 의해 컴퓨터 충돌을 공지 받는다. 따라서, 용어 BSOD(Blue Screen of Death)가 통상적으로 사용된다. 사용자가 BSOD를 받을 때, PC를 다시 시작하는 것 외에 다른 복구 방법이 없다.

[0020] 미디어 서버로서 PC를 사용하는 데 특정한 손해는 여러개의 실행중인 오디오 및 오디오 비주얼 어플리케이션 중 단 하나만이 소프트웨어 버그를 가지고 있을 때 발생한다. 소프트웨어 버그는 PC로부터 렌더링된 오디오에서 여러 레벨의 혼란을 야기할 수 있다. 최소한의 혼란 상태에서는, 버그를 가진 어플리케이션에 의해 발생된 오디오 데이터 스트림만 영향 받는다. 혼란 레벨이 높아지면, 모든 오디오 장치들이 영향을 받고 아마도 동작을 멈춰서, 모든 오디오 장치에 대한 오디오의 글로벌 렌더링이 중단될 것이다. 심지어 더 심한 시나리오에서는, 오디오 어플리케이션들 중 하나에 있는 소프트웨어 버그가 컴퓨터 충돌을 발생시켜 OS가 다시 로드될 때까지 PC에서 모든 활동을 중단하게 한다 - 이것은 몇 분이 걸릴 수 있다.

**발명이 이루고자 하는 기술적 과제**

[0021] 이와같이, 오디오 어플리케이션을 고립시키는 한편, 컴퓨팅 시스템에서 실행되는 어떤 종류의 오디오 어플리케이션의 요청이라도 서비스를 하는 OS를 제공하여, 실행중의 오디오 어플리케이션의 실패가 글로벌 오디오 렌더링을 중단하게 하거나 컴퓨터 충돌을 유발하지 않게 하는 것은 본 기술 분야에서 장점이 될 것이다.

**발명의 구성 및 작용**

[0022] 한 구현에서, 오퍼레이팅 시스템의 시스템 오디오 서비스는 오디오 어플리케이션의 문맥을 보존하며, 오디오 어플리케이션의 프로세싱이 실패하더라도 프로세싱은 복구될 수 있다. 또 다른 구현에서, 컴퓨팅 시스템은 제1 오디오 데이터 스트림을 생성하는 오디오 어플리케이션을 실행하기 위한 프로세서를 갖는다. 프로세서는 커널을 위한 파티션(partition)과 오디오 서브시스템을 위한 파티션으로 논리적으로 분리된 오퍼레이팅 시스템을 실행한다. 오디오 서브시스템은 논리적으로 구분된 서브파티션들로 분리된다. 오디오 서브시스템으로 제1 오디오 데이터 스트림을 처리하기 위한 프로세스(예를 들어, 오디오 프로세싱 오브젝트)는 제2 서브파티션에서 프로세스를 실행하기 전에 제1 서브파티션에서 보존되는 문맥을 갖는다. 프로세스가 액세스 위반을 일으킬 때, 프로세스는 제1 서브파티션으로 부터 그 문맥을 복구함으로써 회복된다. 프로세스를 위해 보존되는 문맥은, 오디오 서브시스템의 보존된 문맥을 복구함으로써 프로세스를 회복시키기 위해 충분한 주소와 데이터 구조 정보를 포함한다. 프로세서가 제1 오디오 데이터 스트림이 출력되는 제1 오디오 장치와 다른 제2 오디오 장치로 출력되는 제2 오디오 데이터 스트림을 생성하기 위해 오디오 어플리케이션을 실행할 때, 액세스 위반과 프로세스의 복구 중 어떤 것도 제2 오디오 장치에 의해 출력되는 오디오에 영향을 주지 않을 것이다.

- [0023] 상세한 설명이 첨부된 도면들을 참조하여 제공되며, 상세한 설명 및 도면 전반에서, 동일한 숫자는 동일한 구성 요소 및 특징을 지칭하도록 사용된다. 100번대 숫자는 도 1에서 최초로 나타나는 특징을 나타내고, 200번대 숫자는 도 2에서 최초로 나타나는 특징을 나타내며, 300번대 숫자는 도 3에서 최초로 나타나는 특징을 나타낸다.
- [0024] 다수의 오디오 장치와 그를 구동하기 위한 어플리케이션 장치들은 미디어 서버로서 사용되는 컴퓨팅 시스템과 통신할 수 있다. 이와 같이, 컴퓨팅 시스템은 동시에 하나 이상의 오디오 장치에 오디오를 보낼 수 있다. 컴퓨팅 시스템의 오퍼레이팅 시스템(OS)은 커널과, 오퍼레이팅 시스템의 많은 서브시스템들 중 하나인 내장형 오디오 시스템을 둘다 가지고 있다. 하나 또는 그 이상의 어플리케이션은, 오디오 서브시스템에게 하나 이상의 오디오 장치로 보내지는 모든 오디오에 대해 글로벌 이펙트(GFX)를 가질 오디오 프로세싱 오브젝트(APO)를 실행 하라고 요청한다. 예를 들어, GFX APO는 잔향 이펙트, 왜곡 이펙트, 스피커 보정 이펙트 또는 다른 디지털 신호 프로세싱(DSP) 이펙트 일 수 있다.
- [0025] 오디오 서브시스템은 OS의 커널을 오디오 장치 드라이버들과 각 서로로부터 고립시킨다. 샌드박싱(sandboxing) 기술인 이러한 고립은 액세스 위반을 일으키는 것 등에 의해 오디오 어플리케이션이 실패할 때, 복구를 허용한다. 그 고립은 실행되고 있는 모든 어플리케이션으로부터의 모든 오디오가 아니라, 실패한 어플리케이션으로부터의 오디오만이 중단되는 것을 보장한다. 또한, 이 고립은 액세스 위반이 컴퓨터 시스템의 충돌을 유발하지 않을 것도 보장하는데, 이것은 부트(예를 들면, 부트스트랩) 동작을 방지한다. 실패한 어플리케이션은 재시작 되어, 오디오를 계속 생성할 수 있다. 어플리케이션의 실패로 인해, 사용자는 단지 순간적인 오디오 손실만을 경험한다. 컴퓨터 시스템의 사용자에게 의해 인지된 오디오 손실은 실패한 어플리케이션의 오디오에 한정되고, 손실된 사운드 기간에 따라 달라질 것이다.
- [0026] 한 구현에서, 2명의 사용자가 미디어 서버로서 퍼스널 컴퓨터(PC)를 사용하고 있을 수 있는데, 이 때 제1 사용자는 PC의 DVD 플레이어로 DVD 상의 영화를 재생하고 있고, 그 영화의 사운드가 제1 스피커 세트에서의 렌더링을 위한 제1 사운드 카드에 의해 처리되고 있다. 제2 사용자는 PC의 CD 리더를 사용하여 콤팩트 디스크(CD) 상의 음악을 재생하고 있다. 제2 사용자에게 관해서는, 재생은 PC의 마더보드와 통신하는 제2 사운드 카드와 통신하는 제2 스피커 세트에 렌더링 된다. 제2 사운드 카드는 오디오 장치 드라이버에 의해 구동된다. 음악의 재생은 PC의 미디어 플레이어 어플리케이션의 실행에 의해 수행된다. 미디어 플레이어 어플리케이션은 음악 재생에 의해 생성된 오디오 데이터 스트림 상에 GFX APO를 호출한다. GFX APO가 액세스 위반을 일으키는 것 등에 의해 실행 동안 실패하게 하는 소프트웨어 버그를 가지고 있는 경우, 제2 사용자는 그 실패로부터 회복하는 동안, 예를 들어, 5초의 사운드 손실을 경험할 것이다. 회복 동안에, 컴퓨터는 다시 시작할 필요가 없고 제1 사용자는 DVD 상의 영화의 재생에서 어떠한 사운드의 손실도 느끼지 않을 것이다. GFX APO가 다시 로드되고, 오디오 서브시스템은 음악의 재생을 위해 오디오를 멈춰 있었던 곳으로 되돌린다. 다시 로드된 GFX APO는 제2 사운드 카드로 제2 오디오 세트에 렌더링된 오디오 데이터 스트림을 처리하기 위해 실행을 계속한다. 어떤 다른 오디오 장치가 PC를 사용하여 오디오를 렌더링하고 있는 중이었던 경우, 그 오디오는 액세스 위반이나 회복에 의해서 손실되지 않을 것이다.
- [0027] 도 1은 PC(124)가 미디어 서버로서 사용되는 환경(100)을 보여준다. 많은 사람들이 도 1에 보여지며, 이들 각각은 하나 이상의 스피커(106)등으로부터 렌더링된 오디오를 듣고 있다. PC(124)는 미디어 서버로서 PC(124)를 사용하고 있는 사람들의 이익을 위해 오디오를 포함하는 멀티미디어를 재생하기 위하여, 디지털 비디오 디스크(DVD) 리더 및 콤팩트 디스크(CD) 리더와 같은 휴대용 미디어 리더들을 A-V 섹션(122) 내에 가지고 있다. PC(124)는 A-V 섹션(122) 내의 휴대용 미디어 리더들로부터 관독되고 있는 휴대용 미디어를 재생하기 위해 복수의 멀티미디어 어플리케이션을 실행한다. 오디오 데이터는 오퍼레이팅 시스템(114)과 통신하는 입력 섹션(120)과 통신하고 있다. 이러한 멀티미디어 어플리케이션 중 하나는 USA 워싱턴 레드몬드의 마이크로소프트 사에 의해 제공되는 Window? Media Player 소프트웨어일 수 있다.
- [0028] PC(124)의 O/S(114)는 각각에 대응하는 드라이버(104)(1-N)를 통해 여러가지 오디오 장치(102)(1-N)와 통신하는 출력 섹션(112)을 갖는다. 예를 들어, 오디오 장치(102)는 2개의 채널 사운드 카드, 한 쌍의 Universal Serial Bus(USB) 스피커, 네트워크 스피커(TCP/IP 스피커), 또는 하나 이상의 사운드 채널의 출력을 수용할 수 있는 또 다른 장치 등일 수 있다. 예를 들어, 하나의 오디오 장치(102)는, PC(124)의 마더 보드에 장착되어 디지털 데이터를 아날로그 사운드로 변환함으로써 PC(124)가 스피커(106)들 중 하나에 사운드를 렌더링할 수 있도록 하는 확장 보드인 사운드 카드가 될 수 있다. 오디오 장치(102)는 오디오 데이터를 렌더링(재생)하거나 캡처(녹음)하는 데 사용될 수 있다. 예를 들어, 오디오 장치(102)는 PC(124)에 연결된 마이크로폰(106)(k-1)으로부터 입력된 사운드를 기록하는 데 사용될 수 있고, 디스크에 저장된 사운드를 처리하는 데 사용될 수 있다. 한 사람이 하나 이상의 스피커(106)(k)를 들으면서 마이크로폰(106)(k-1)에 말하고 있는 것으로 도시된다. 오디오는

O/S(114)의 동작을 통해 렌더링되고 캡처된다. 마이크론(106)(k-1)은, 한 사람이 마이크론(106)(k-1)에 말하여 네트워크를 통해 다른 PC의 사용자와 말할 수 있게 네트워크 전화 통신 환경에서 스피커(106)(K)와 함께 사용될 수 있으며, 이 때 각 PC는 리얼 타임 통신 어플리케이션(RTC)을 실행한다. O/S(114)는 논리적으로 구분된 파티션들인 오디오 서브시스템(116)과 커널(118)을 포함한다. 커널(118)은 오퍼레이팅 시스템(114)을 위한 스케줄링 서비스이다. 오디오 컴포넌트(116)는 도 2와 관련하여 아래에 설명하는 것과 같은 다양한 소프트웨어 컴포넌트를 포함할 수 있다.

[0029] 도 1의 O/S(114)의 구현이 도 2에서 보여지는데, 여기에서 환경(200)은 사운드가 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치에 의해 렌더링(render)될 수 있도록 한다. 도 2의 굵고 진한 선은, O/S(114)가 선(220)에 의해 정의되는 논리적으로 구분된 2개의 파티션을 갖고, 오디오 서브시스템(116)이 굵은 선(220-224)에 의해 정의되는 논리적으로 구분된 3개의 서브파티션을 가진다는 것을 나타내기 위한 것이다. O/S(114)의 논리적으로 구분된 2개의 파티션은 선(220)에 의해 나뉘어진 커널(118)과 오디오 서브시스템(116)이다. 각 파티션은 사용자 모드에서 자신의 주소 공간을 가지며, 서로에 대해, 및 실행되고 있는 임의의 어플리케이션에 대해 완전히 고립되어 있다. 마찬가지로, 각 서브파티션은 사용자 모드에서 자신의 주소 공간을 갖고, 그것은 완전히 서로, 및 실행되고 있는 다른 어플리케이션으로부터 고립된다. 3개의 서브파티션은 모두 사용자 모드에서 커널(118)의 최상위에 위치하여, 어느 한 서브파티션이 충돌하는 경우, PC (124)가 치명적인 에러없이 계속 동작하게 한다. 여기서 사용되는 '문맥(context)'이라는 용어는, 프로세스 및 그 프로세스에 관련된 데이터 구조에 관련된 메모리의 주소 공간을 의미하며, 데이터 구조는 프로세스가 적절히 실행되기 위해 오퍼레이팅 시스템에 의해 요구된다.

[0030] 환경(200)은 오디오 서브시스템(116)을 제어하는 명령의 흐름을 반영한 실선 화살표를 갖는다. 환경(200)에서 점선 화살표는 오디오 데이터 스트림의 흐름을 나타낸다. 오디오 서브 시스템(116) 내의 논리적으로 구분된 3개의 서브파티션은 프로세스 문맥(270), 서비스 문맥(272), 및 글로벌 오디오 엔진(274)이다. 글로벌 오디오 엔진(274)은 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치 하드웨어와 동기화될 수 있다. 동기화될 때, 글로벌 오디오 엔진(274)은 적당한 시기에 데이터를 루프된 버퍼(212)로 펌프한다. 동기화 된 펌핑은 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치가 적당한 시기에 루프된 버퍼(212)로부터 오디오 데이터 스트림을 읽을 수 있게 한다.

[0031] 오디오 어플리케이션(202)은 프로세스 문맥(270)에서 실행된다. 프로세스 문맥(270)은 오디오 서브시스템(116)에서 실행중인 각 오디오 어플리케이션(202)을 위해 구분된 주소 공간을 갖는다. 이와 같이, 각 오디오 어플리케이션(202)의 실행은 다른 오디오 어플리케이션(도시 안됨)에 영향을 주지 않을 것이다. 프로세스 문맥(270)은 오디오 어플리케이션(202)을 위해 오디오 데이터 스트림의 처리를 수행한다. 오디오 어플리케이션(202)의 소프트웨어에서의 에러는, 오디오 어플리케이션(202)에 의해 각 웨이크업 기간 동안에 수행되는 오디오 데이터 프로세싱의 용량, 스피드 및 주파수 때문에 자주 발생할 수 있다.

[0032] 선(220-222)에 의해 정의된 프로세스 문맥(270)은 서비스 문맥(272)에 의해 서비스 또는 관리된다. 프로세스 문맥(270)은 일반적인 어플리케이션 프로세스 문맥에서 실행되는 클라이언트(또는 어플리케이션) 공간을 대표한다. 오디오 어플리케이션(202)은 프로세스 문맥(270)에서 로컬 이펙트 엔진(LFX)에 의해 처리될 수 있는 오디오 데이터 스트림을 생성하며, 그 오디오 데이터 스트림은 입력 버퍼(208)로 출력된다. 입력 버퍼(208)는 프로세스 문맥(270)과 글로벌 오디오 엔진(274)의 사이에 존재한다. 달리 말하자면, 입력 버퍼(208)는 프로세스 문맥(270)에서 동작하는 로컬 어플리케이션으로부터의 출력을 저장한다. 도 2는 프로세스 문맥(270)에 대하여 하나의 오디오 어플리케이션(202)만을 나타내고있지만, 오디오 서브시스템(116)은 서브시스템(116)의 프로세스 문맥(270)에 대하여 복수의 오디오 어플리케이션을 지원할 수 있을 것으로 예상된다.

[0033] 서비스 문맥(272)은 오디오 서브시스템(116)에서 선(222-224)에 의해 정의된다. 서비스 문맥(272)은 PC(124)의 부팅시에 O/S(114)가 로드할 때, 또는 사용자가 로그인 또는 로그오프를 할 때 존재하는 오디오 서브시스템(116)의 컴포넌트이다. 서비스 문맥(272)은 어떤 오디오 어플리케이션이 오디오 서브시스템 (116)의 서비스로 액세스하기 위해 요구하는 어플리케이션 프로그램 인터페이스(API)를 포함한다. 서비스 문맥(272)은 시스템 오디오 서비스에서 실행되는 정책(policy) 및 시스템 그래프 서비스와 같은 글로벌 오디오 서비스 공간을 대표한다. 서비스 문맥(272)은 프로세스 문맥(270)에 의한 프로세싱의 충돌 후에 남아있다.

[0034] 글로벌 오디오 엔진(274)은, 글로벌 오디오 엔진(274)이 루프된 버퍼(212)를 통해 통신하는 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치(도시 안됨, 예를들어, 사운드 카드)를 위한 프락시(proxy)이다. 글로벌 오디오 엔진(274)은 각 글로벌 오디오 엔진 프로세싱 공간마다 하나의 프로세스를 나타내며, 각 글로벌 오디오

엔진 프로세싱 공간은 어떤 하나의 특정 오디오 장치를 위한 모든 오디오 데이터의 처리를 담당한다. 추가적으로, 오디오 서브시스템(116)의 글로벌 오디오 엔진(274)은 오디오 데이터 스트림을 생성하는 각각의 오디오 또는 멀티미디어 어플리케이션에 대한 복수의 접속을 혼합함으로써 단일의 오디오 장치를 가상화(virtualizing)할 수 있다. 이와 같이 가상화되면, 글로벌 오디오 엔진(274)은 혼합된 오디오 데이터 스트림들을 하나의 오디오 데이터 스트림으로서 단일 오디오 장치에 제공한다. 이와 같이, 동시에 실행되는 다수의 어플리케이션은 그들 모두가 공유하는 동일한 오디오 장치로 출력할 수 있다. 추가적으로, 글로벌 오디오 엔진(274)은 다수의 오디오 장치가 PC(124)가 미디어 서버로 사용되는 동안 동시에 기능할 수 있도록 할 수 있다.

[0035] 입력 버퍼(208) 내의 오디오 데이터 스트림은 글로벌 오디오 엔진(274)에 대한 입력으로서 사용될 수 있다. 글로벌 오디오 엔진(274)은 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치에서 출력되는 오디오 데이터 스트림에 글로벌 이펙트(GFX)를 가지는 오디오 프로세싱 오브젝트(APO)를 수행한다. 이와 같이, 글로벌 오디오 엔진(274)은 입력 버퍼(208) 내의 오디오 데이터 스트림을 혼합하거나 처리할 수 있는데, 여기서 입력 버퍼(208)는 복수의 오디오 어플리케이션으로부터의 오디오 데이터 스트림을 저장하는 데 사용된다. 입력 버퍼(208) 내의 오디오 데이터 스트림들을 혼합하고 처리하는 것의 결과는 하나의 혼합된 오디오 데이터 스트림의 루프된 버퍼(212)로의 출력이다. 루프된 버퍼(212)는 오디오 장치 드라이버(214)로의 입력을 위해 글로벌 오디오 엔진(274)과 인터페이스한다. 이와 같이, 루프된 버퍼(212)는 글로벌 오디오 엔진(274)으로부터의 출력을 저장하는데 사용될 수 있고, 오디오 장치 드라이버(214)로의 입력으로서 사용될 수 있다.

[0036] 환경(200)의 오디오 서브시스템(116)은 각 글로벌 오디오 엔진(274)마다 하나의 프로세스를 제공한다. 각 글로벌 오디오 엔진(274)은 대응하는 오디오 장치 드라이버와의 인터페이스를 통해 각 오디오 장치에 대한 모든 오디오 데이터 스트림의 프로세싱을 순서대로 담당한다. 도 2가 하나의 오디오 장치에 대하여 단 하나의 오디오 장치 드라이버(214)만을 도시하고 있지만, 오디오 서브시스템(116)은 복수의 오디오 장치와 그들 각각의 오디오 장치 드라이버들을 지원할 수 있다는 것이 예상된다.

[0037] 전술한 것과 관련하여, 오디오 어플리케이션(202), 글로벌 오디오 엔진(274)에 의해 실행되는 GFX APO, 오디오 장치 드라이버(214) 각각은, 오디오 장치에서 오디오 데이터 스트림을 렌더링하기 위하여 환경(200)의 오디오 서브시스템(116)에 의해 사용되도록 제3자에 의해 제공될 수 있는 소프트웨어를 나타낸다. 이와 같이, 환경(200)은, 제3자들이 환경(200)의 오디오 서브시스템(116)이 상주하는 PC(124)를 사용하는 고객을 위하여 오디오 소프트웨어를 혁신하고 제공할 장소와 마켓을 제공한다.

[0038] 도 2에 도시된 환경(200)은 O/S(114)의 컴포넌트를 불완전한 제3자 소프트웨어의 실행에 의해 발생하는 액세스 위반으로 인한 문제들로부터 고립시킨다. 샌드박싱(sandboxing)으로도 알려진 이러한 고립은 오디오 데이터 스트림이 오디오 서브시스템(116)의 서브파티션들(270 및 274)에서 처리되는 문맥을 보존함으로써 이루어진다. 한 구현예에서, 글로벌 오디오 엔진(274)의 문맥만이 서비스 문맥(272)에 의해 보존된다. 액세스 위반으로 인해 발생할 수 있는 것과 같이 글로벌 오디오 엔진(274)이 충돌하는 경우, 충돌은 프로세스 문맥(270)이나 오디오 어플리케이션(202)에 영향을 주지 않을 것이다. 오디오 어플리케이션(202)이 충돌하는 경우, 오디오 서브시스템(116) 내의 어떤 것도 영향을 받지 않을 것이다. 서비스 문맥(272)이 충돌하는 경우, 전체 오디오 서브시스템(116)이 충돌할 것이다.

[0039] 글로벌 오디오 엔진의 회복

[0040] 글로벌 오디오 엔진(274)에서 GFX APO의 프로세싱에 액세스 위반에 의한 것과 같은 실패가 있다면, 서비스 문맥(272) 내에 보존되었던 이 프로세싱의 문맥은 그 프로세싱을 복구하는 데 사용될 수 있다. 복구 후에, 오디오 데이터 스트림의 프로세싱은, 서비스 문맥(272) 내에 보존되어 있는 대로, 프로세스의 보존된 문맥을 따라 다시 시작할 수 있다. 오디오 서브시스템(116)에 의해 수행된 오디오 요구들을 서비스하는 것은 커널(276)로부터 샌드박싱된다. 이 샌드박싱은 시스템 전반의 실패를 방지한다. 로컬 및 글로벌 프로세싱 어플리케이션 실패들로 인한 것을 포함하여 충돌하거나 실패하는 어플리케이션들은 프로세싱이 다시 시작할 수 있도록 자동적으로 복구될 수 있다.

[0041] 로컬 및 글로벌 양자 모두의 프로세싱 문맥의 보존은 프로세싱의 메커니즘을 자신의 주소 공간에 놓음으로써 달성될 수 있다. 달리 말하자면, 프로세스 문맥(270) 내에 발생하는 로컬 오디오 엔진의 프로세싱 루틴의 문맥은, 글로벌 오디오 엔진(274)의 프로세싱 루틴의 문맥을 저장한 주소와 구분된 주소에 저장된다. 프로세스의 주소 공간은 프로세스의 문맥이 보존되는 주소와 구분되어 있다. 프로세싱의 메커니즘이 문맥의 메커니즘의 주소와 구분된 자신의 주소 공간에 놓여질 때, 문맥의 메커니즘이 손상되지 않고 남아 있는 한편, 프로세싱의 메커니즘은 충돌할 수 있다. 오디오 서브시스템(116)과 논리적으로 구분된 커널(118) 또한 손상되지 않고 남아

있다. 이와 같이, 충돌은 하나의 오디오 어플리케이션을 위한 로컬 프로세싱 또는 하나의 오디오 장치를 위한 글로벌 프로세싱의 종료만으로 한정될 것이다. 충돌은 PC(124)의 실패를 유발하지 않을 것이며, 따라서 그렇지 않을 경우에는 오디오의 렌더링을 다시 시작하기 위해 요구되었을 시스템 재시작이나 재부팅을 방지한다. 충돌 후에, 오디오의 렌더링을 다시 시작하기 위하여, PC(124)의 O/S(114)를 다시 시작하기 보다는, 프로세싱의 메커니즘이 다시 시작될 수 있다. 어떤 예들은 실패된 프로세싱이 다시 시작될 수 있도록 문맥이 보존될 것으로 예상된다.

[0042] 이하에서는, 오디오 데이터 스트림의 글로벌 오디오 프로세싱에서의 실패로부터의 복구에 관한 예가, 미디어 플레이어 어플리케이션이 오디오 데이터를 재생하는 구현의 일례로서 주어진다. 오디오 데이터는 버그를 갖는 제3자 소프트웨어 오디오 프로세싱 오브젝트(APO)를 사용하는 글로벌 오디오 이펙트(GFX)에 영향을 받고 있다. GFX APO 소프트웨어는 글로벌 오디오 엔진(274)에서 오디오 데이터 스트림을 처리하는 디지털 신호 프로세싱 어플리케이션이다. 환경(200)은 글로벌 오디오 엔진(274)에서 발생 중인 프로세싱의 문맥의 서비스 문맥(272)에서 보존(예를 들면, 저장)을 조정한다. 이와 같이, 글로벌 오디오 엔진(274) 내의 GFX APO를 위한 주소 공간과 데이터 구조들은 서비스 문맥(272)에 저장된다. 결국, 글로벌 오디오 엔진(274)에서 실행하는 GFX APO 내의 버그는 액세스 위반을 일으킨다. 이에 따라, 액세스 위반은 글로벌 오디오 엔진(274) 내의 프로세싱이 충돌하게 한다. 충돌의 결과는 오디오 장치 드라이버(214)를 통해 오디오 장치 상에서 렌더링되고 있던 오디오가 일정 기간 동안 멈추는 것이다. 이 기간 동안, 환경(200)은 충돌로부터의 회복을 조정한다.

[0043] 회복은 글로벌 오디오 엔진(274) 내의 GFX APO의 프로세싱의 문맥을 복구함으로써 수행된다. 문맥은 그 문맥이 보존되었던 서비스 문맥(272)으로부터 회복된다. 따라서, 복구되는 보존된 문맥은 입력 버퍼(208)의 문맥, 글로벌 오디오 엔진(274) 내의 GFX APO 프로세싱의 문맥, 및 루프된 버퍼(212)의 문맥이다. 일단 서비스 문맥(272)으로부터 각 문맥이 복구되면, 글로벌 오디오 엔진(274) 내의 GFX APO 프로세싱은 계속 진행할 수 있다. 문맥 복구 후에, 오디오는 계속해서 오디오 장치 드라이버(214)를 통해 오디오 장치 상에 렌더링될 수 있다. 컴퓨팅 시스템을 사용하는 고객은, GFX APO가 글로벌 오디오 엔진(274) 내의 보존된 문맥의 복구에 의해 다시 로드되는 동안, 짧은(예를 들어, 5초) 손실된 오디오를 듣게 될 것이다. 그런 경우에 오디오는 대략 그것이 끊긴 곳으로 복귀하고 다시 시작한다. 환경(200)에 의해 각 오디오 장치의 오디오가 다른 오디오 장치들의 오디오로부터 고립되기 때문에, 다른 오디오 장치들 중 어떤 것도 GFX APO의 충돌에 의해 오디오 손실을 입지 않는다.

[0044] 충돌에서 회복되는 동안, 글로벌 오디오 엔진(274)에 의한 오디오 데이터의 프로세싱은 오디오 장치 드라이버(214)에 의해 서비스 되고 있는 오디오 장치로의/로부터의 모든 오디오 스트리밍을 멈추게 할 것이다. 그러나, 각 오디오 어플리케이션(202)은 충돌이 발생했다는 것을 알지 못할 것이다. 오디오 장치의 오디오 그래프 구조를 유지하는 것을 담당하는 서비스 문맥(272)이 별개의 서비스 문맥 내에서 실행되기 때문에, 서비스 문맥(272)은 이전의 그 보존으로 인해, 충돌 후에 글로벌 오디오 엔진(274)의 문맥을 복구하는 데 사용될 수 있다. 서비스 문맥(272)은 또한 오디오 장치 드라이버(214)의 상태(즉, 시작한 것인지 또는 멈춘 것인지)와 다른 프로세스 문맥들을 식별하는 서브파티션들 사이에서 사용되는 공유된 버퍼들[예를 들어, 입력 버퍼(208), 루프된 버퍼(212)]을 제어한다. 서비스 문맥(272)은 입력 버퍼(208), 루프된 버퍼(212) 및 글로벌 오디오 엔진(274)을 그들 각각의 서브파티션 내에서 다시 연결할 것이다. 오디오 어플리케이션(202)은 계속 진행되고, 충돌에 투명성을 가질 것이다(예를 들어, 충돌 또는 액세스 위반이 일어나지 않은 것처럼).

[0045] 한 구현에서, 서비스 문맥(272)은 임의의 제3자 GFX APO 소프트웨어가 충돌한 횟수를 추적할 수 있다. 그 횟수가 미리 정해진 임계값을 초과한 후, 사용자에게 대한 진단 메시지 또는 다른 대화가 사용자 인터페이스(UI)에 출력될 수 있다. GFX APO도 사용할 수 없게 되거나, 사용자의 UI에 경고가 출력된 후에 다시 사용 가능하게 될 수 있다. 진단 메시지가 UI에 출력되고 제3자 GFX APO 소프트웨어의 실행이 글로벌 오디오 엔진(274)에서 다시 시작되지 않는 것과 같이, 다수의 액세스 위반과 그 후의 회복에 대한 다른 "단계적인" 접근들이 행해질 수 있다. 액세스 위반 후의 복구에 대한 단계적인 접근은 손상된 글로벌 이펙트 컴포넌트의 브링 업(bring up) 및 테이크 다운(take down)의 지나친 반복을 피하는 데 좋은 영향을 준다.

[0046] 오디오 어플리케이션(202)에 대해서, 글로벌 오디오 엔진(274)에 의한 오디오 데이터 프로세싱의 충돌은 오디오 어플리케이션(202)의 수행과 실행에 투명하다. 오히려, 오디오 어플리케이션(202)은 프로세스 문맥(270)과 글로벌 오디오 엔진(274) 사이의 파티션 경계에 있는 입력 버퍼(208)로의 출력을 위해 프로세스 문맥(270)에 계속해서 입력한다. 서비스 문맥(272)은 오디오 어플리케이션(202)이 GFX APO 충돌을 인식하지 못하고 있는 동안, 문맥을 복구하고 글로벌 오디오 엔진(274)의 자동적인 회복을 수행하는 데 사용될 수 있다.

- [0047] 버퍼(208, 212)는 글로벌 오디오 엔진(274)에서의 프로세싱이 충돌한 후에도 여전히 존재하기 때문에, 오디오 장치 드라이버(214)의 상태와 루프된 버퍼(212)는 복구될 수 있다. 글로벌 오디오 엔진(274) 내의 프로세싱이 충돌한 후에 서비스 문맥(272)이 그 프로세싱을 셋업할 때, 새로운 글로벌 오디오 엔진(274)이 버퍼(208, 212)에 접속하고, 그렇지 않으면 글로벌 오디오 엔진(274)의 충돌이나 버퍼(208, 212)의 주소를 인식하지 못하는 오디오 어플리케이션(202)으로부터 계속해서 오디오 데이터를 받을 수 있도록, 서비스 문맥(272)은 버퍼(208, 212)의 주소를 알아야 한다. 오히려, 서비스 문맥(272)이 글로벌 오디오 엔진(274)을 회복시킬 때, 버퍼(208, 212)가 글로벌 오디오 엔진(274) 내에 셋업될 수 있도록, 서비스 문맥(272)은 버퍼(208, 212)를 오브젝트로서 유지한다. 서비스 문맥(272)은, 버퍼(208, 212)가 특정 오디오 장치를 위한 오디오 어플리케이션(202)로부터의 입력으로서 사용될 것을 지시하는 명령들을 글로벌 오디오 엔진(274)에 전달한다. 글로벌 오디오 엔진(274)이 충돌할 때, 복구된 글로벌 오디오 엔진(274)은 서비스 문맥(272)으로부터 루프된 버퍼(212)의 주소를 얻는다. 액세스 위반 후에, 오디오 장치 드라이버(214)는 여전히 실행 중일 수 있고, 루프된 버퍼(212)는 그대로 존재할 수 있다.
- [0048] 앞에서 언급한 것은 서비스 문맥(272) 내의 문맥의 보존으로 인한 글로벌 오디오 엔진(274) 내의 충돌된 GFX APO로부터의 자동적인 회복의 예를 보여준다. 서비스 문맥(272) 내의 문맥의 보존은, 단 하나의 오디오 장치에 대한 오디오가 PC(124)를 다시 시작하는 데 요구되는 시간에 비해 짧은 시간 동안만 중단될 것을 보장한다. 소프트웨어 개발자들은 본 명세서에 개시된 오디오 서브시스템(116)에서 실행되는 GFX APO를 제공할 수 있으며, GFX APO는 여전히 모든 오디오 장치들을 파멸시키거나 OS(114) 전체를 충돌시키지 않고 충돌할 수 있다. GFX APO는 문맥이 서비스 문맥(272) 내에 보존되는 한 되돌려질 수 있다. 오디오 서브시스템(116)은 각 오디오 장치(예를 들어, 사운드 카드, USB 스피커 등)에 대응하는 글로벌 오디오 엔진(274)을 제공하여, 글로벌 오디오 엔진(274)이 자신의 문맥에서 실행되고, 서비스 문맥(272)에서 보존될 수 있도록 할 수 있다. 이와 같이, 오디오 서브시스템(116)은 장애를 견딜 수 있고, 충돌/속력 저하에 강하고, 글로벌 오디오 엔진(274)에 의한 GFX APO 내의 오디오 데이터의 프로세싱의 충돌로부터 회복할 수 있다.
- [0049] 각 오디오 장치는 환경(200)에서 보여지는 구조에 따라 구분된 프로세스 문맥으로 나뉘어질 수 있다. 이와 같이, 하나의 특정 오디오 장치를 위한 글로벌 오디오 엔진(274) 내의 프로세싱에 의해 글로벌 오디오 이펙트로서 실행 중인 충돌하기 쉬운 소프트웨어는, 다른 오디오 장치를 위한 글로벌 오디오 엔진의 오디오 데이터의 프로세싱에 영향을 주지 않을 것이다. 게다가, 프로세싱 중인 글로벌 이펙트 코드의 버그로 인해 글로벌 오디오 엔진(274)에서의 프로세싱이 충돌할 때, 자동적인 회복이 달성될 수 있다. 오디오 장치의 글로벌 상태가 서비스 문맥(272)에 저장되므로 회복이 가능하다. 이와 같이, 충돌된 GFX APO는 되돌려지고 입력 버퍼(208)에 다시 연결될 수 있다.
- [0050] 일반적으로, 글로벌 오디오 엔진(274) 내의 프로세싱에서의 충돌로부터의 회복은, 오디오를 오디오 장치를 위해 오디오 데이터 스트림을 생성하는 각각의 어플리케이션으로부터 대응하는 오디오 장치로 가져간다. 가능한 한 충돌에 저항하기 위해, 각 오디오 장치를 위한 글로벌 오디오 엔진(274)은 자신의 문맥에서 실행될 수 있으며, 그 문맥은 오디오 서브시스템(116)의 논리적으로 구분된 서비스 문맥(272)에서 보존된다.
- [0051] 도 3은 글로벌 오디오 엔진 내에서 실행하는 GFX APO에 의한 액세스 위반으로부터 회복하는 프로세스(300)를 보여준다. 도 2-3을 참고하면, 프로세스(300)는 서비스 문맥(272)이 입력 버퍼(208)와 루프된 버퍼(212)를 위한 문맥(예를 들어, 주소와 데이터 구조)을 보존하는 블록(302)에서 시작한다. 블록(304)에서, 오디오 데이터 스트림 상에 GFX를 생성하기 위한 APO가 시작된다. 오디오 데이터 스트림은 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치에 의해 렌더링된다. 프로세스(300)는 서비스 문맥(272)이 글로벌 오디오 엔진(274) 내의 GFX APO를 위한 문맥을 보존하는 블록(306)으로 이동한다. 블록(306) 후에, 프로세스(300)는 GFX APO가 시작되는 블록(308)으로 이동한다. 글로벌 오디오 엔진(274) 내에서의 실행은, 글로벌 오디오 엔진(274)이 입력 버퍼(208) 내의 오디오 데이터 스트림에 대한 디지털 신호 프로세싱을 수행하기 위하여 GFX APO를 실행하는 블록(310)에서 계속 진행된다. 다른 로컬 프로세스들로부터의 혼합된 오디오 데이터 스트림들을 포함할 수 있는 이 프로세싱의 결과는 루프된 버퍼(212)로 이동된다. 루프된 버퍼(212) 내의 혼합된 오디오 데이터 스트림은 글로벌 오디오 엔진(274)에 의한 프로세싱에 따른 글로벌 이펙트를 갖는다.
- [0052] 블록(312)에서, GFX APO는 종료하거나 액세스 위반을 유발한다. 블록(314)에서, 서비스 문맥(272)은 종료 및 액세스 위반을 검색하고, 입력 버퍼(208) 및 루프된 버퍼(212)의 문맥을 복구한다. 블록(316)에서, 액세스 위반이 발생했는지의 여부에 관한 질의가 만들어진다. 발생하지 않은 경우, 프로세스(300)는 블록(304)으로 이동한다. 발생한 경우, 프로세스(300)은 GFX APO를 위한 충돌 카운터가 증가되는 블록(318)으로 이동한다. 충돌 카운터는 블록(320)에서 미리 정해진 임계값을 초과하는지에 대한 질문을 받는다. 초과하는 경우, 진단 메세지

(322a)가 디스플레이되고, 프로세스(300)는 글로벌 오디오 엔진(274)이 충돌하기 쉬운 GFX APO 없이 다시 시작되는 블록(322b)으로 이동한다. GFX APO를 위한 충돌 카운터가 미리 정해진 임계값을 초과하지 않는 경우, 프로세스(300)는 블록(308)으로 되돌아간다.

[0053] 프로세스 문맥의 회복

[0054] 오디오 데이터 스트림의 로컬 프로세싱의 실패로부터의 회복이 이루어질 수 있다. 예를 들어, 도 2에서 오디오 어플리케이션(202)으로서 보여지는 미디어 플레이어 어플리케이션은 프로세스 문맥(270)에서 로컬 이펙트 오디오 프로세싱 오브젝트(LFX APO)에 의한 프로세싱을 위해 오디오 데이터 스트림을 통과시킨다. LFX APO는 실행되었을 때 액세스 위반(예를 들어, 코드에 버그가 있음)을 유발하는 프로그래밍 코드의 결함 부분을 갖는다. 프로세스 문맥(270)에서 LFX APO를 실행시키기 전에, 환경(200)의 오디오 서브시스템(116)은 프로세스 문맥(270)에서 발생 중인 프로세싱 문맥에 대한 서비스 문맥(272) 내에서의 보존(예를 들면, 저장)을 조정한다. 결국, LFX APO의 버그는 소프트웨어가 액세스 위반을 수행하게 한다. 액세스 위반은 차례로 프로세스 문맥(270) 내의 프로세싱이 충돌하게 한다. 충돌의 결과는 입력 버퍼(208)로 렌더링되고 있던 오디오가 일정 기간 동안 멈춘다는 것이다. 이와 같이, 오디오 장치 드라이버(214)를 통해 오디오 장치에 렌더링되는 오디오에 공백이 생기게 될 것이다. 이 기간 동안 환경(200)은 충돌로부터의 회복을 조정한다.

[0055] 프로세스 문맥(270) 내의 LFX APO의 충돌로부터의 회복은 프로세스 문맥(270) 내의 프로세싱의 문맥을 복구함으로써 수행된다. 일단, 문맥이 서비스 문맥(272)으로부터 복구되면, 오디오 어플리케이션(202)의 실행이 다시 시작될 수 있어서, 프로세스 문맥(270) 내의 LFX APO 프로세싱이 계속 진행될 수 있게 된다. 일단 로컬 프로세싱이 계속 진행되면, 오디오 데이터 스트림은 입력 버퍼(280)로 출력될 수 있고, 사운드는 오디오 장치 드라이버(214)의 동작을 통해 오디오 장치에서 다시 시작할 수 있다. 오디오 어플리케이션(202)이 환경(200)에 의해 다른 오디오 어플리케이션들로부터 고립되어 있기 때문에, 어떤 다른 오디오 어플리케이션도 충돌하지 않을 것이다.

[0056] 프로세싱 문맥(270)에 의한 LFX APO의 프로세싱에서의 충돌은 입력 버퍼(208)를 오디오 어플리케이션(202)으로부터 분리시킬 것이다. 프로세싱 문맥(270)의 프로세싱에서의 충돌 후에 남아있는 임의의 오디오 데이터 스트림은 클린업될 필요가 있을 것이다. 충돌이 발생할 때, 서비스 문맥(272)은 충돌을 통보받을 수 있다. 이와 같이 통보받으면, 서비스 문맥(272)은 충돌 후에 남아있는 데이터 스트림을 닫을 수 있다. 남아있는 오디오 데이터 스트림을 클린업하기 위해, 서비스 문맥(272)은 프로세스 문맥(270)에 의한 오디오 데이터의 프로세싱을 위해 만들어진 리스트를 유지할 수 있다. 이 리스트는 남아있는 오디오 데이터 스트림을 갖고 있는 프로세스 문맥(270)의 종료 (또는 다른 그러한 프로세스 문맥의 종료)를 기다리는 쓰레드를 남기기 위해 서비스 문맥(272)에 의해 사용될 수 있다.

[0057] 프로세스 문맥(270)에 의한 프로세싱이 충돌하는 경우, 서비스 문맥(272)은 멈추고 프로세스 문맥(270)에 의한 프로세싱으로부터 열려 있는 모든 오디오 데이터 스트림들을 닫을 것이다. 오디오 어플리케이션(202)이 충돌하는 경우, 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치 또는 글로벌 오디오 엔진(274) 어떤 것도 영향을 받지 않을 것이다. 오히려, 오디오 데이터는 다른 어플리케이션들에 의해 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치 또는 글로벌 오디오 엔진(274)으로 스트리밍될 수 있다. 그러나 프로세스 문맥(270)에서의 충돌 때문에, 입력 버퍼(208)는 프로세스 문맥(270)으로부터 분리된다. 이와 같이, 입력 버퍼(208)의 오디오 데이터는 사용될 수 없는 것이다. 서비스 문맥(272)이 프로세스 문맥(272)에 의한 프로세싱이 충돌한 것을 탐지하면, 입력 버퍼(208)로의 각 연결이 단히거나 할당 해제된다. 한 구현에서, 오디오 어플리케이션(202)은 자동적으로 회복되지 않는다. 오히려, 사용자가 희망한다면, 오디오 어플리케이션(202)은 사용자에 의해 다시 시작될 수 있다.

[0058] 도 4는 프로세스 문맥(270)에서 실행되는 로컬 이펙트(LFX) 오디오 프로세싱 오브젝트(APO)에 의한 액세스 위반으로부터 회복하는 프로세스(400)를 보여준다. 도 2 및 도 4를 참고하면, 프로세스(400)는 오디오 데이터 스트림 상에 LFX를 생성하기 위한 APO가 시작 되는 블록(402)에서 시작한다. 오디오 데이터 스트림은 오디오 장치 드라이버(214)에 의해 구동되는 오디오 장치에 의해 렌더링된다. 블록(404)에서, 서비스 문맥(272)은 입력 버퍼(208) 및 루프된 버퍼(212)의 문맥(예를 들어, 주소 및 데이터 구조)을 보존한다. 블록(406)에서, LFX APO가 프로세스 문맥(270)에서 실행을 시작한다. 블록(408)에서, 프로세스 문맥(270)은 입력 버퍼(208)로 출력하고, 글로벌 오디오 엔진(274)이 입력 버퍼(208)로부터 입력을 받는다. 글로벌 오디오 엔진(274)은 입력 버퍼(208)로부터의 오디오 데이터 스트림들을 혼합하여, 그 혼합된 오디오 데이터 스트림을 루프된 버퍼(212)에 출력한다. 블록(410)에서, LFX APO가 종료되거나 액세스 위반을 유발한다. 블록(412)에서, 서비스 문맥(272)이

종료 또는 액세스 위반을 탐지하여, 입력 버퍼(208)를 삭제한다.

[0059] 각 컴퓨팅 시스템을 위한 다양한 오퍼레이팅 시스템들은 앞에서 언급한 구현들, 특히 오퍼레이팅 시스템이 오디오 데이터의 캡처와 렌더링의 결합을 위한 오디오 서비스를 제공하는 부분, 및 오퍼레이팅 시스템이 멀티 쓰레드와 멀티 테스킹이라는 부분으로부터 이익을 얻을 수 있다. 예시적인 최신의 오퍼레이팅 시스템들은 미국 캘리포니아 쿠파ertino 소재의 애플 컴퓨터에서 제공하는 맥 OS(버전 10 이하), 캘리포니아 마운틴 뷰 소재의 비(Be, Inc.)가 개발한 BeOS, 미국 워싱턴 레드몬드 소재의 마이크로소프트사가 제공하는 Window? OS, 프리웨어 리눅스 OS를 포함한다. 게다가, 본 명세서로부터의 이익은, 신뢰할 수 없는 미디어 어플리케이션이 액세스 위반을 수행하거나 오퍼레이팅 시스템이 빠르게 동작하는 것을 중단시킬 수 있는 제3자 프로그램을 실행하는 것을 허용하는 멀티 쓰레드 오퍼레이팅 시스템에 의해 실현될 수 있다. 예를 들어, 여기서 예상되는 오퍼레이팅 시스템은 오디오 신호를 변경하기 위해 글로벌 문맥에서 사용될 수 있는 오디오 프로세싱 이팩트를 위하여 신뢰할 수 없는 제3자 미디어 어플리케이션에 대해 확장 가능한 미디어 프로세싱 시스템을 제공한다. 불충분하게 기록된 코드 또는 악성 코드는 특정 오디오 장치를 서비스하고 있는 오퍼레이팅 시스템의 오디오 시스템의 일부를 충돌시키지 않고 실행될 수 있다. 프로세스와 같이, 오디오 서브시스템을 오퍼레이팅 시스템 내의 보존 메커니즘을 사용하는 논리적으로 구분된 파티션들로 분할함으로써, 다양한 구현에서 충돌 방지가 달성된다. 그 프로세스는 불충분하게 썩여진 코드가 오퍼레이팅 시스템에 주는 영향을 최소화한다. 프로세스에 관련된 주소 공간과 데이터 구조들은 프로세스가 실행되도록 하기 위하여 오퍼레이팅 시스템에 의해 사용된다. 이와 같이, 오퍼레이팅 시스템은 메모리 보호의 한 형태 및 프로세스(또는, 쓰레드)로 대표될 수 있는 문맥 보존의 한 형태로서 고립된 프로세싱 문맥에서 프로세스를 실행시킨다.

[0060] 예시적인 컴퓨팅 시스템 및 환경

[0061] 도 5는 리얼 타임 커뮤니케이션 어플리케이션(RTC)과 미디어 플레이어 어플리케이션(MP)을 포함한 여기서 설명된 어플리케이션들이 완전히 또는 부분적으로 구현될 수 있는 컴퓨팅 환경(500)의 예를 나타낸다. 예시적인 컴퓨팅 환경(500)은 컴퓨팅 시스템의 일례일 뿐이며, 네트워크 구조의 사용 또는 기능의 범위에 대해 어떠한 제한도 제안하고자 하는 것은 아니다. 또한, 컴퓨팅 환경(500)은 예시적인 컴퓨팅 환경(500)에 도시된 컴포넌트들 중의 하나 또는 그 조합과 관련해서 어떠한 종속성이나 필요 조건을 갖는 것으로 해석되어서도 안된다.

[0062] 컴퓨터와 네트워크 아키텍처는 다수의 다른 범용 또는 전용 컴퓨팅 시스템 환경 또는 구성으로 구현될 수 있다. 사용에 적합한 잘 알려진 컴퓨팅 시스템, 환경 및/또는 구성의 예는, 퍼스널 컴퓨터, 서버 컴퓨터, 썬 클라이언트(thin client), 씨크 클라이언트(thick client), 휴대형 또는 랩탑 장치, 멀티 프로세서 시스템, 마이크로 프로세서 기반 시스템, 셋 탑 박스, 프로그램가능한 소비자 전자 장치, 네트워크 PC, 미니 컴퓨터, 메인 프레임 컴퓨터, 게임 콘솔, 상기 시스템 또는 장치 중 임의의 것을 포함하는 분산 컴퓨팅 환경등을 포함하지만, 이들로 제한되는 것은 아니다.

[0063] RTC 및 MP를 포함하는 어플리케이션은 프로그램 모듈과 같이 컴퓨터에 의해 실행되는 컴퓨터 실행가능한 명령의 일반적인 문맥으로 기술될 수 있다. 일반적으로, 프로그램 모듈은 특정 태스크를 수행하거나 특정 추상 데이터 타입을 구현하는 루틴, 프로그램, 오브젝트, 컴포넌트, 데이터 구조 등을 포함한다. RTC 및 MP를 포함하는 어플리케이션은, 통신 네트워크를 통해 링크된 원격 프로세싱 장치에 의해 태스크가 수행되는 분산 컴퓨팅 환경에서도 실행될 수 있다. 분산 컴퓨팅 환경에서, 프로그램 모듈은 메모리 저장 장치를 포함하는 로컬 및 원격 컴퓨터 저장 장치 매체에 위치될 수 있다.

[0064] 컴퓨팅 환경(500)은 컴퓨터(502)의 형태로 범용 컴퓨팅 시스템을 포함한다. 컴퓨터(502)의 컴포넌트는 하나 이상의 프로세서, 즉 프로세서(504), 시스템 메모리(506), 및 프로세서(504)를 포함한 다양한 시스템 컴포넌트를 시스템 메모리(506)에 연결시키는 시스템 버스(508)를 포함할 수 있지만, 이들로 제한되는 것은 아니다.

[0065] 시스템 버스(508)는 메모리 버스 또는 메모리 컨트롤러, 주변 버스, 가속된 그래픽 포트, 및 다양한 버스 구조 중 임의의 것을 사용하는 프로세서 또는 로컬 버스를 포함한 여러 종류의 버스 구조 중 하나 이상을 나타낸다. 예를 들어, 그러한 구조들은 ISA(Industry Standard Architecture) 버스, MCA(Micro Channel Architecture) 버스, EISA(Enhanced ISA) 버스, VESA(Video Electronics Standards Association) 로컬 버스, 및 메자닌(Mezzanine) 버스로도 알려진 PCI(Peripheral Component Interconnects) 버스를 포함할 수 있다.

[0066] 컴퓨터 시스템(502)은 일반적으로 다양한 컴퓨터 관독가능 매체를 포함한다. 그러한 매체는 컴퓨터(502)에 의해 액세스 될 수 있는 임의의 사용 가능한 매체일 수 있으며, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 포함한다. 시스템 메모리(506)는 랜덤 액세스 메모리(RAM)(510)와 같은 휘발성 메모리 및/또는 관독 전용

메모리(ROM)(512)와 같은 비휘발성 메모리 형태의 컴퓨터 판독가능 매체를 포함한다. 시작되는 동안과 같이, 정보가 컴퓨터(502) 내의 요소들 사이에서 전달되는 것을 돕는 기본 루틴들을 포함하는 기본 입출력 시스템(BIOS)(514)은 ROM(512)에 저장된다. RAM(510)은 기본적으로 즉시 액세스 가능하고/가능하거나 프로세싱 유닛(504)에 의해 현재 수행되고 있는 데이터 및/또는 프로그램 모듈을 포함한다.

[0067] 컴퓨터(502)는 다른 분리형/비분리형, 휘발성/비휘발성 컴퓨터 저장 매체도 포함할 수 있다. 예를 들어, 도 5는 비분리형의 비휘발성 자기 매체(도시되지 않음)에 대한 판독 및 기입을 행하는 하드 디스크 드라이브(516), 분리형의 비휘발성 자기 디스크(520)(예를 들면, 플로피 디스크)에 대한 판독 및 기입을 행하는 자기 디스크 드라이브(518), 및 CD-ROM, DVD-ROM 또는 기타 광 디스크와 같은 분리형의 비휘발성 광 디스크(524)에 대한 판독 및 기입을 행하는 광 디스크 드라이브(522)를 나타낸다. 하드 디스크 드라이브(516), 자기 디스크 드라이브(518) 및 광 디스크 드라이브(522)는 각각 하나 이상의 데이터 매체 인터페이스(525)에 의해 시스템 버스(508)에 연결된다. 다르게는, 하드 디스크 드라이브(516), 자기 디스크 드라이브(518) 및 광 디스크 드라이브(522)는 SCSI 인터페이스(도시되지 않음)에 의해 시스템 버스(508)에 연결된다.

[0068] 디스크 드라이브 및 그 관련 컴퓨터 판독가능 매체는 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈 및 컴퓨터(502)의 다른 데이터의 비휘발성 저장 장치를 제공한다. 본 예는 하드 디스크(516), 분리형 자기 디스크(520) 및 분리형 광 디스크(524)를 나타내고 있지만, 자기 카세트 또는 기타 자기 저장 장치, 플래시 메모리 카드, CD-ROM, DVD 또는 기타 광 저장 장치, RAM, ROM, EEPROM(electronic erasable programmable read-only memory) 등과 같이, 컴퓨터에 의해 액세스 가능한 데이터를 저장하는 다른 종류의 컴퓨터 판독가능 매체도 예시적인 컴퓨팅 시스템 및 환경을 구현하는 데 이용될 수 있음을 알것이다.

[0069] 예를 들어, 오퍼레이팅 시스템(526), 하나 이상의 어플리케이션 프로그램들(528), 기타 프로그램 모듈(530) 및 프로그램 데이터(532)를 포함하는 임의의 개수의 프로그램 모듈이 하드 디스크(516), 자기 디스크(520), 광 디스크(524), ROM(512) 및 RAM(510)에 저장될 수 있다. 이러한 오퍼레이팅 시스템(526), 하나 이상의 어플리케이션 프로그램(528), 다른 프로그램 모듈(530), 프로그램 데이터(532) 각각(또는 이들의 결합)은 본 명세서에 개시된 RTC 또는 MP의 실시예를 포함할 수 있다. 미디어 어플리케이션이 신호 프로세싱을 위한 광범위한 혼합인 글로벌 오디오 출력 시스템을 얻을 수 있도록, 오퍼레이팅 시스템(526)은 오디오 캡처 및 렌더링 서비스를 포함할 수 있다.

[0070] 컴퓨터 시스템(502)은 통신 매체로 식별되는 다양한 컴퓨터 판독가능 매체를 포함할 수 있다. 통신 매체는 일반적으로 컴퓨터 판독가능 명령, 데이터 구조, 프로그램 모듈, 또는 반송파와 같이 변조된 데이터 신호 또는 기타 전송 매커니즘 내의 다른 데이터를 구현하고, 정보 전달 매체를 포함한다.

[0071] 용어 "변조된 데이터 신호"은 신호 내의 정보를 인코딩하는 것과 같은 방식으로 설정 또는 변경된 하나 이상의 특징을 갖는 신호를 의미한다. 예를 들어, 통신 매체는 유선 네트워크 또는 직통 유선 접속과 같은 유선 매체, 및 음향, RF, 적외선 또는 기타 무선 매체와 같은 무선 매체를 포함하지만, 이들로 제한되는 것은 아니다. 이들의 임의의 조합도 컴퓨터 판독가능 매체의 범위 내에 포함된다.

[0072] 사용자는 키보드(534) 및 포인팅 장치(536)(예를 들면, 마우스)와 같은 입력 장치를 통해서 명령 및 정보를 컴퓨터 시스템(502)에 입력할 수 있다. 마이크로폰(535)은 음성 입력에 전달하기 위해서 음성 인식 프로세스될 수 있는 음성 명령을 입력하는 데 사용될 수 있다. 다른 입력 장치(538)(도시되지 않음)는 조이스틱, 게임 패드, 위성 접시, 직렬 포트, 스캐너 및/또는 그와 유사한 것들을 포함할 수 있다. 여기에 개시된 것과 그 외의 입력 장치들은 시스템 버스(580)에 결합된 입출력 인터페이스(540)를 통해 프로세싱 유닛(504)에 접속되지만, 병렬 포트, 게임 포트 또는 USB와 같은 기타 인터페이스 및 버스 구조도 접속될 수 있다.

[0073] 모니터(542)나 다른 종류의 디스플레이 장치도 비디오 어댑터(544)와 같은 인터페이스를 통해 시스템 버스(508)에 접속될 수 있다. 입출력 인터페이스(540)는 사운드 카드, 일체형(예를 들면, 온-보드형) 사운드 카드 등을 포함할 수 있다. 하나 이상의 스피커(537)가 입출력 인터페이스(540)와 통신할 수 있다. 모니터(542) 이외에 다른 출력 주변 장치로는, 입출력 인터페이스(540)를 통해 컴퓨터(502)에 접속될 수 있는 프린터(546)와 같은 컴포넌트가 포함될 수 있다.

[0074] 컴퓨터(502)는 원격 컴퓨팅 장치(548)와 같은 하나 이상의 원격 컴퓨터들과의 로컬 연결을 이용하여 네트워크화된 환경에서 동작할 수 있다. 예를 들어, 원격 컴퓨팅 장치(548)는 퍼스널 컴퓨터, 휴대용 컴퓨터, 서버, 라우터, 네트워크 컴퓨터, 피어 장치, 다른 일반적인 네트워크 노드 등일 수 있다. 원격 컴퓨팅 장치(548)는 여기서 컴퓨터 시스템(502)과 관련되어 설명된 많은 또는 모든 요소와 특성이 포함될 수 있는 휴대용 컴퓨터로서 설

명된다.

[0075] 컴퓨터(502)와 원격 컴퓨터(548) 간의 논리적인 연결은 근거리 네트워크(LAN)(550)와 일반적인 광역 네트워크(WAN)(522)로 묘사된다. 그러한 네트워킹 환경들은 사무실, 기업형 컴퓨터 네트워크, 인트라넷 및 인터넷에서 흔히 볼 수 있다. LAN 네트워킹 환경에서 구현될 때, 컴퓨터(502)는 네트워크 인터페이스 또는 어댑터(554)를 통해 근거리 네트워크(550)에 연결된다. WAN 네트워킹 환경에서 구현될 때, 컴퓨터(502)는 광역 네트워크(552) 상에서의 통신을 위해 일반적으로 모뎀(556) 또는 다른 수단을 포함한다. 컴퓨터(502)에 내장 혹은 외장될 수 있는 모뎀(556)은 입출력 인터페이스(540) 또는 다른 적당한 메커니즘을 통해 시스템 버스(508)로 연결될 수 있다. 설명된 네트워크 연결들은 예시적이며, 컴퓨터들(502 및 548) 간의 통신 링크를 확립하기 위한 다른 수단이 사용될 수 있다는 것을 알 것이다.

[0076] 컴퓨팅 환경(500)으로 나타낸 것과 같은 네트워크화된 환경에서, 컴퓨터(502)와 관련되어 설명된 프로그램 모듈 또는 그 부분들은 것들은 원격 메모리 저장 장치에 저장된다. 예를 들어, 원격 어플리케이션 프로그램들(558)은 원격 컴퓨터(548)의 메모리 장치 상에 상주한다. 설명의 목적으로서, 어플리케이션 프로그램 및 오퍼레이팅 시스템과 같은 다른 실행가능한 프로그램 컴포넌트가 개별적인 블록으로서 설명되지만, 이러한 프로그램과 컴포넌트는 다양한 시간에 컴퓨터 시스템(502)의 다른 저장 장치들에 상주하며, 컴퓨터의 데이터 프로세서에 의해 실행된다는 것을 알 것이다.

[0077] 본 발명이 구조적인 특징 및/또는 방법론적인 단계에 특수한 언어로 설명되었지만, 첨부된 특허청구범위에서 정의된 발명은 반드시 상기에 설명된 특정 특징이나 단계로 제한되지 않는다는 것을 알 것이다. 반대로, 특정 특징과 단계는 청구된 발명을 구현하는 바람직한 형태로서 개시된다.

**발명의 효과**

[0078] 본 발명에 따르면, 실행 동안의 오디오 어플리케이션의 실패로 인해, 글로벌 오디오 렌더링이 중단되거나 컴퓨터 충돌이 발생하지 않는다.

**도면의 간단한 설명**

[0001] 도 1은, 컴퓨팅 시스템이 복수의 장소에서 복수의 사용자에게 오디오를 서비스하기 위한 미디어 서버로서 사용되는 환경에서의 예시적인 컴퓨팅 시스템의 도면.

[0002] 도 2는, 오디오 장치를 위한 드라이버와 통신하는 오퍼레이팅 시스템을 가지고 있는 컴퓨팅 시스템에서 오디오 어플리케이션이 실행하고 있는 구현을 나타낸 도면.

[0003] 도 3은, 오퍼레이팅 시스템의 오디오 서브시스템이 오디오 프로세싱 오브젝트(APO)를 실행하기 위한 요청을 서비스하여, 오디오 드라이버에 의해 구동되는 오디오 장치에 의해 렌더링되는 오디오 데이터 스트림에 대한 글로벌 이펙트(GFX)를 생성하는 프로세스를 설명하는 플로우 차트로서, GFX APO에 의한 액세스 위반 후, 여러가지 오디오 서브시스템의 문맥은 이전에 복구된 문맥을 사용하여 복구됨.

[0004] 도 4는, 오퍼레이팅 시스템의 오디오 서브시스템이 오디오 프로세싱 오브젝트(APO)를 실행하기 위한 요청을 서비스하여, 오디오 드라이버에 의해 구동되는 오디오 장치에 의해 렌더링되는 오디오 데이터 스트림에 대한 로컬 오디오 이펙트(LFX)를 생성하는 프로세스를 설명하는 플로우 차트로서, LFX APO에 의한 액세스 위반 후, 여러가지 오디오 서브시스템의 문맥은 이전에 복구된 문맥을 사용하여 복구됨.

[0005] 도 5는 본 명세서에 설명된 컴퓨팅 시스템, 소프트웨어 어플리케이션, 방법 및 시스템이 전체적 또는 부분적으로 구현될 수 있는 컴퓨팅 환경의 일례를 나타낸 도면.

[0006] <도면의 주요 부분에 대한 부호의 설명>

[0007] 116 : 오디오 서브시스템

[0008] 118 : 커널

[0009] 202 : 오디오 어플리케이션

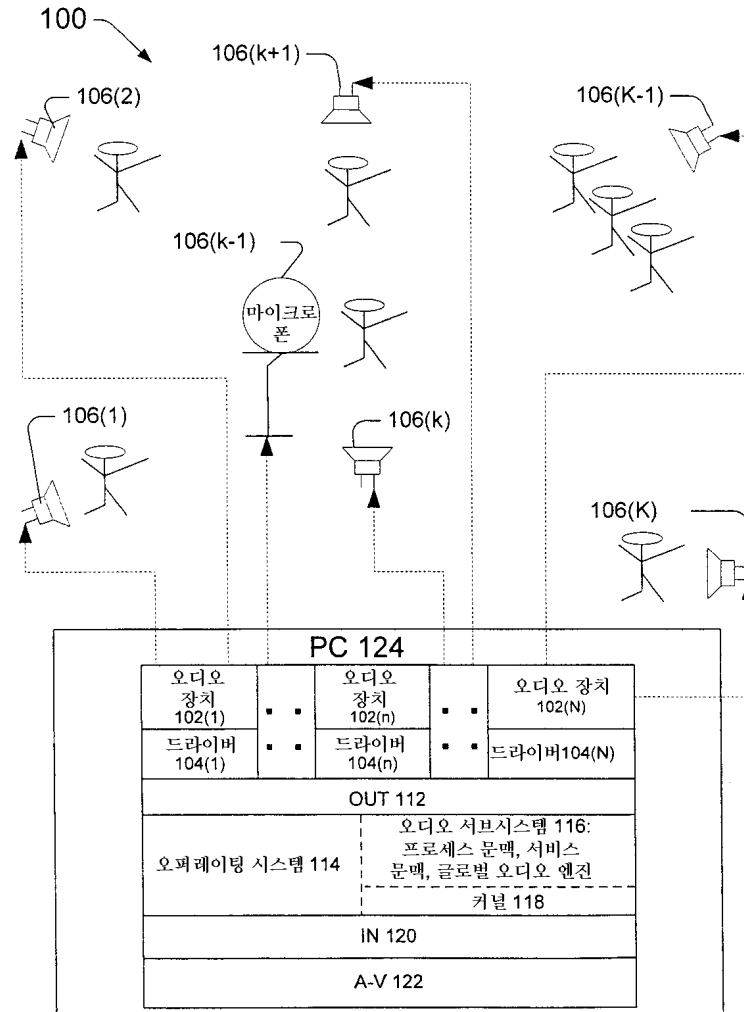
[0010] 208 : 입력 버퍼

[0011] 212 : 루프된 버퍼

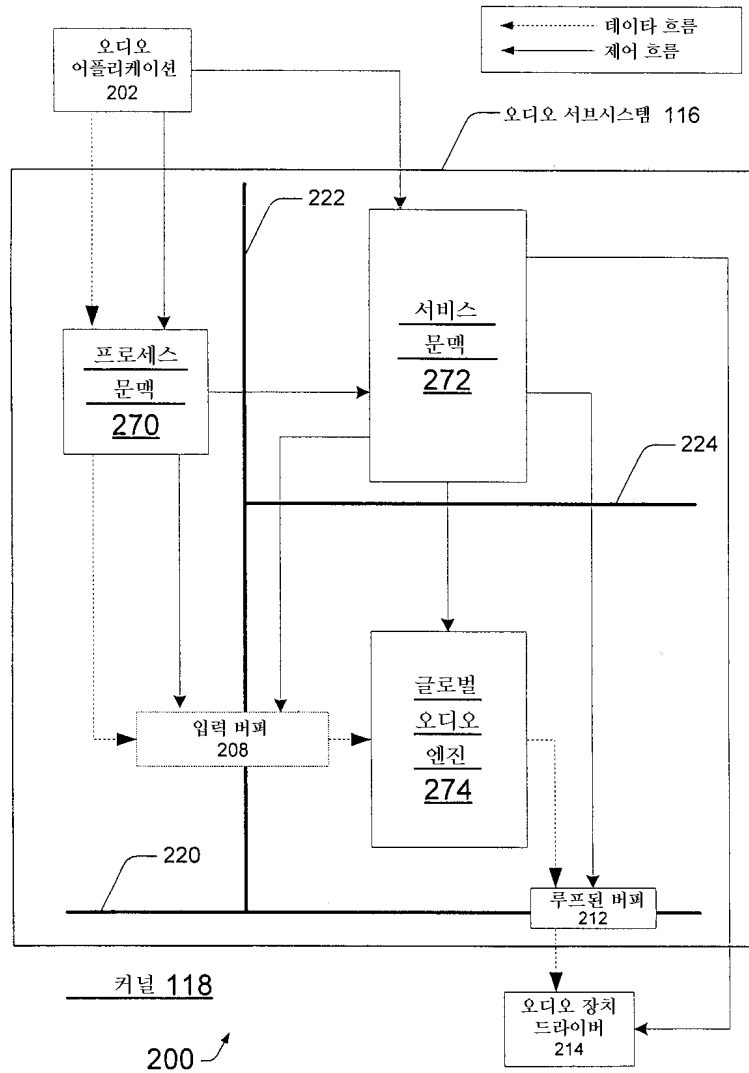
- [0012] 214 : 오디오 장치 드라이버
- [0013] 270 : 프로세스 문맥
- [0014] 272 : 서비스 문맥
- [0015] 274 : 글로벌 오디오 엔진

도면

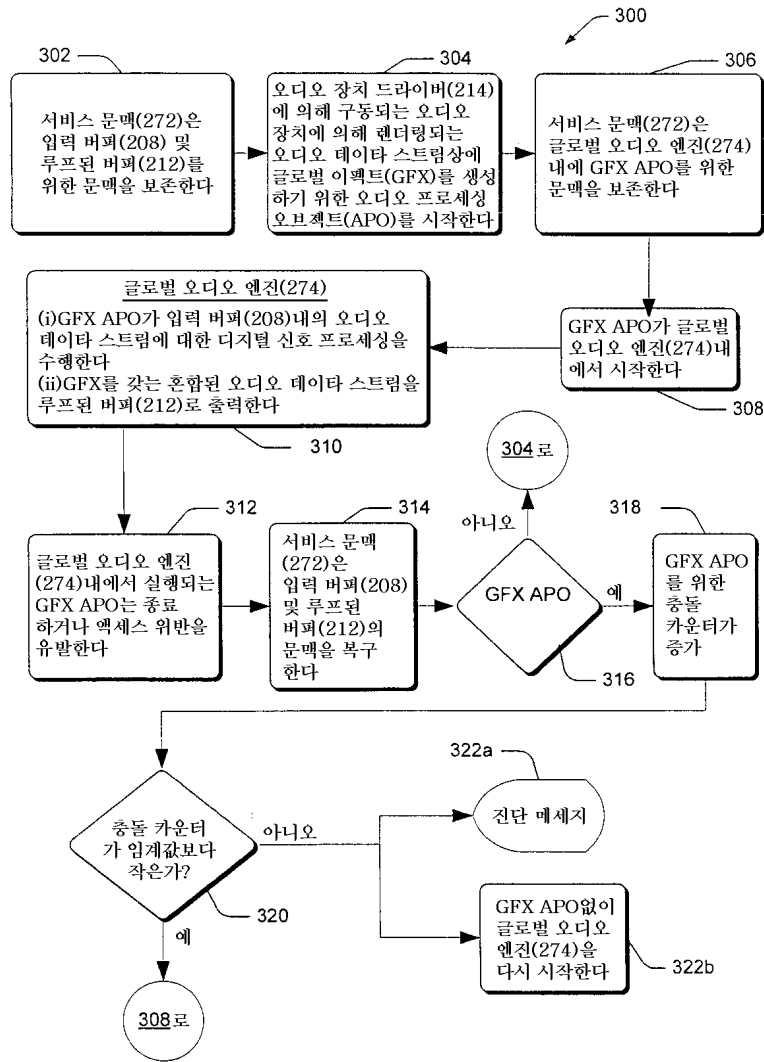
도면1



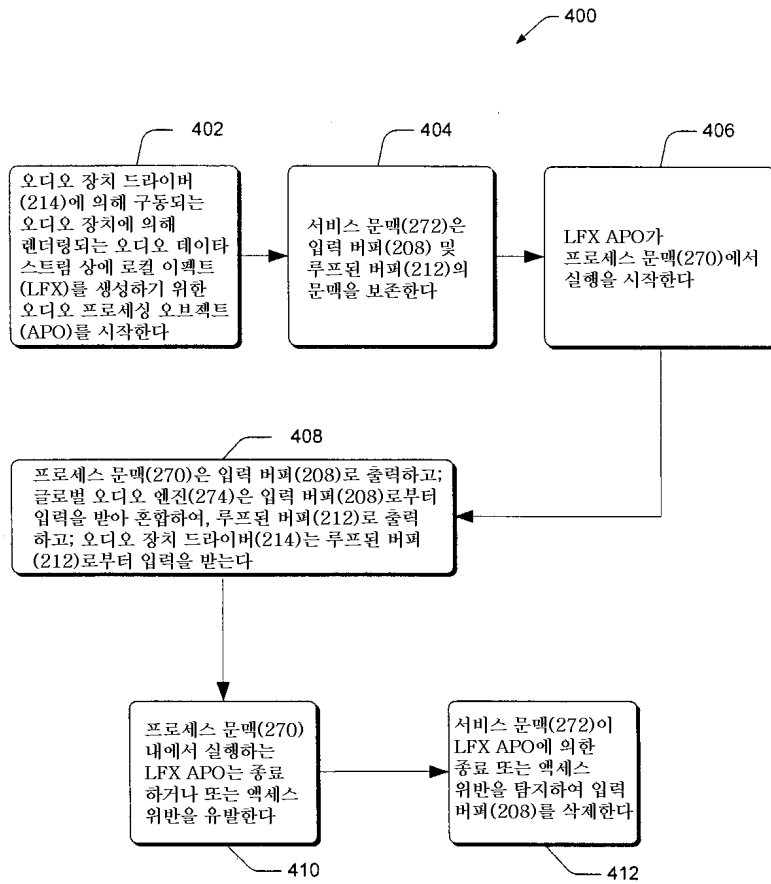
도면2



도면3



도면4



도면5

