



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0138146
(43) 공개일자 2013년12월18일

(51) 국제특허분류(Int. Cl.) <i>G06F 9/46</i> (2006.01) <i>G06F 9/38</i> (2006.01)	(71) 출원인 애플 인크. 미합중국 95014 캘리포니아 쿠퍼티노 인피니트 루프 1
(21) 출원번호 10-2013-0065533	(72) 발명자 라메시, 아난스아크리시나 미국 95014 캘리포니아주 쿠퍼티노 엠/에스 3-씨 오에스 인피니트 루프 1
(22) 출원일자 2013년06월07일 심사청구일자 2013년06월07일	린, 요센 미국 95014 캘리포니아주 쿠퍼티노 엠/에스 302-3오에스3 인피니트 루프 1 (뒷면에 계속)
(30) 우선권주장 13/656,506 2012년10월19일 미국(US) 61/657,611 2012년06월08일 미국(US)	(74) 대리인 백만기, 양영준

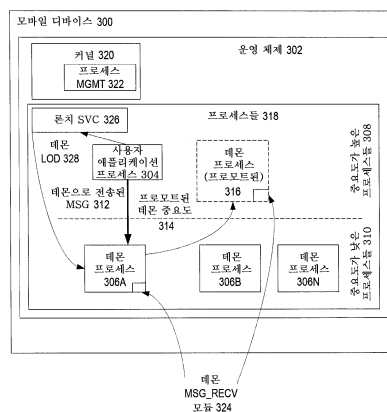
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **적용형 프로세스 중요도**

(57) 요약

데몬 프로세스의 중요도를 변경하는 디바이스의 방법 및 장치에 대해 설명된다. 예시적인 실시예에서, 디바이스는 데몬 프로세스로 향하는 사용자 프로세스로부터의 메시지를 수신하는데, 데몬 프로세스는 사용자 프로세스와 독립적으로 실행하고, 제1 데몬 프로세스는 다른 실행 프로세스들과 메시지들을 통신한다. 디바이스는, 제1 메시지가 제1 데몬 프로세스의 중요도가 변경될 수 있음을 나타내는지를 더 결정한다. 제1 메시지가, 제1 데몬 프로세스의 중요도가 변경될 수 있음을 나타내는 경우, 디바이스는 제1 데몬 프로세스의 중요도를 변경한다. 디바이스는 또한 제1 데몬 프로세스에 제1 메시지를 포워드한다.

대표도 - 도3a



(72) 발명자

소레스, 다미엔 파스칼

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 3-씨오
에스 인피니트 루프 1

마지, 제임스 마이클

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 111-에
이치오엠 인피니트 루프 1

반 베치텐, 케빈 제임스

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 3-씨오
에스 인피니트 루프 1

마이삼파얀, 우메시 스투시

미국 95014 캘리포니아주 쿠퍼티노 엠/에스 3-씨오
에스 인피니트 루프 1

특허청구의 범위

청구항 1

하나 이상의 처리 유닛이 제1 데몬 프로세스의 중요도를 변경하는 방법을 수행하게 하는 실행가능한 명령어들을 갖는 비-일시적인 기계-판독가능한 매체로서,

상기 방법은,

상기 제1 데몬 프로세스로 향하는 사용자 프로세스로부터의 제1 메시지를 수신하는 단계 - 상기 제1 데몬 프로세스는 상기 사용자 프로세스와 독립적으로 실행하고, 상기 제1 데몬 프로세스는 다른 실행 프로세스와 메시지들을 통신하고, 상기 메시지는 상기 사용자 프로세스로부터 상기 제1 데몬 프로세스로의 서비스 요청임 - ;

상기 제1 메시지가 상기 제1 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는지 결정하는 단계;

상기 제1 메시지가 상기 제1 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는 경우, 상기 제1 데몬 프로세스의 상기 중요도를 변경하는 단계; 및

상기 제1 데몬 프로세스로 상기 제1 메시지를 포워드하는 단계

를 포함하는 비-일시적인 기계-판독가능한 매체.

청구항 2

제1항에 있어서, 상기 사용자 프로세스는 상기 제1 데몬 프로세스보다 더 높은 중요도로 실행되는 비-일시적인 기계-판독가능한 매체.

청구항 3

제1항에 있어서, 상기 제1 데몬 프로세스의 상기 중요도를 변경하는 단계는, 상기 제1 데몬 프로세스의 중요도를 상기 사용자 프로세스와 동일한 중요도로 변경하는 비-일시적인 기계-판독가능한 매체.

청구항 4

제1항에 있어서, 상기 결정하는 단계는,

상기 메시지가 제1 데몬 프로세스의 중요도가 변경될 수 있는 제1 데몬 프로세스로 향하는지를 결정하는 단계를 포함하는 비-일시적인 기계-판독가능한 매체.

청구항 5

제1항에 있어서, 상기 메시지를 포워드하는 단계는,

상기 제1 데몬 프로세스를 위한 포트 큐(port queue)에 상기 메시지를 인큐하는 단계를 포함하는 비-일시적인 기계-판독가능한 매체.

청구항 6

제1항에 있어서,

상기 제1 데몬 프로세스에 대한 어썬션들(assertions)의 카운트를 유지하는 단계;

상기 제1 데몬 프로세스에 대한 드롭 어썬션을 수신하는 단계;

상기 제1 데몬 프로세스에 대한 상기 어썬션들의 카운트를 감소시키는 단계; 및

상기 제1 데몬 프로세스에 대한 상기 어썬션들의 카운트가 제로인 경우, 상기 제1 데몬 프로세스의 상기 중요도를 디폴트 중요도로 변경하는 단계

를 더 포함하는 비-일시적인 기계-판독가능한 매체.

청구항 7

제1항에 있어서,

제2 데몬 프로세스로 향하는 사용자 프로세스로부터의 제2 메시지를 수신하는 단계 - 상기 제2 데몬 프로세스는 실행중이고, 상기 제2 데몬 프로세스는 상기 제1 데몬 프로세스와 독립적으로 실행하고, 상기 제2 데몬 프로세스는 다른 실행 프로세스들과 메시지들을 통신함 - ;

상기 제2 메시지가 상기 제2 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는지를 결정하는 단계;

상기 제2 메시지가 상기 제2 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는 경우, 상기 제2 데몬 프로세스의 상기 중요도를 변경하는 단계; 및

상기 제2 데몬 프로세스로 상기 제2 메시지를 포워드하는 단계

를 더 포함하는 비-일시적인 기계-판독가능한 매체.

청구항 8

제7항에 있어서, 상기 제2 데몬 프로세스의 상기 중요도를 변경하는 단계는, 상기 제2 데몬 프로세스의 상기 중요도를 상기 사용자 프로세스와 동일한 중요도로 변경되도록 변경하는 비-일시적인 기계-판독가능한 매체.

청구항 9

제1항에 있어서,

상기 제1 메시지를, 중요도를 운반하는 것으로서 마크하는 단계

를 더 포함하는 비-일시적인 기계-판독가능한 매체.

청구항 10

제1항에 있어서, 상기 제1 데몬으로 상기 제1 메시지를 포워드하는 단계는 상기 제1 데몬 프로세스 중요도를 변경하는 단계에 대해 응답하는 단계인 비-일시적인 기계-판독가능한 매체.

청구항 11

제1항에 있어서, 상기 사용자 프로세스는 포어그라운드(foreground)에서 실행중인 사용자 애플리케이션을 위한 프로세스인 비-일시적인 기계-판독가능한 매체.

청구항 12

제1항에 있어서, 상기 제1 데몬 프로세스는 백그라운드(background)에서 실행중인 비-일시적인 기계-판독가능한 매체.

청구항 13

제1항에 있어서, 프로세스 중요도는 사용자를 대신하여(on behalf of) 액션이 필요하다는 것을 나타내는 비-일시적인 기계-판독가능한 매체.

청구항 14

하나 이상의 처리 유닛이 적응형 데몬 프로세스의 중요도를 변경하는 방법을 수행하게 하는 실행가능한 명령어들을 갖는 비-일시적인 기계-판독가능한 매체로서,

상기 방법은,

적응형 데몬으로 향하는 중요도 제공자 프로세스(importance donor process)로부터의 메시지를 수신하는 단계 - 상기 적응형 데몬은 아직 론치되지 않음 - ;

상기 메시지가 상기 적응형 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는지를 결정하는 단계;

상기 메시지가 상기 적응형 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는 경우, 상기 적응형 데몬

프로세스와 관련된 디폴트 중요도와는 다른 중요도로 상기 적응형 데몬 프로세스를 론치하는 단계; 및
 론치된 상기 적응형 데몬 프로세스로 상기 메시지를 포워드하는 단계
 를 포함하는 비-일시적인 기계-관독가능한 매체.

청구항 15

제14항에 있어서, 상기 적응형 데몬 프로세스는 변경가능한 프로세스 중요도를 갖는 데몬 프로세스인 비-일시적인 기계-관독가능한 매체.

청구항 16

제14항에 있어서, 상기 중요도 제공자 프로세스는 또 다른 프로세스에 중요도를 제공할 수 있는 프로세스인 비-일시적인 기계-관독가능한 매체.

청구항 17

제16항에 있어서, 중요도 제공자 프로세스는 사용자 애플리케이션 프로세스와, 그의 중요도가 프로모트된 또 다른 적응형 데몬 프로세스로 구성된 그룹으로부터 선택되는 비-일시적인 기계-관독가능한 매체.

청구항 18

제14항에 있어서, 상기 적응형 데몬 프로세스 중요도는 상기 중요도 제공자의 중요도로 프로모트되는 비-일시적인 기계-관독가능한 매체.

청구항 19

제14항에 있어서, 상기 메시지는 상기 중요도 제공자 프로세스로부터 상기 적응형 데몬 프로세스로의 서비스 요청인 비-일시적인 기계-관독가능한 매체.

청구항 20

제1 데몬 프로세스의 중요도를 변경하는 장치로서,
 제1 메시지가 상기 제1 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는지를 결정하기 위한 수단;
 상기 제1 메시지가 상기 제1 데몬 프로세스의 상기 중요도가 변경될 수 있음을 나타내는 경우, 상기 제1 데몬 프로세스의 상기 중요도를 변경하기 위한 수단; 및
 상기 제1 데몬 프로세스로 상기 제1 메시지를 포워드하기 위한 수단을 포함하는 장치.

명세서

기술분야

[0001] 출원인은, 2012년 6월 8일 출원되고, 그 전체 내용이 참조로 통합된, 선행의 동시 가출원 제61/657,611호의 이득을 우선권으로 주장한다.

[0002] 본 발명은 일반적으로 프로세스 관리에 관한 것으로, 더 구체적으로는, 수신된 메시지에 기초하여 프로세스 중요도를 수정하는 것에 관한 것이다.

배경기술

[0003] 모바일 디바이스는, 사용자와 상호작용하고, 그 모바일 디바이스에서 다양한 서비스를 수행하는데 사용되는 복수의 실행중인 프로세스를 갖는 디바이스이다. 예를 들어, 모바일 디바이스는 포어그라운드(foreground)에서 실행중인 하나의 사용자 애플리케이션 프로세스를 가질 수 있고, 여기서, 사용자는 이 프로세스와 상호작용할 수 있다. 실행중인 다른 프로세스들은 사용자 애플리케이션 프로세스 또는 실행중인 다른 프로세스들에게 서비스를 제공하는 데몬 프로세스들이다. 이러한 다른 프로세스들은, CPU(central process units), 메모리, 입/출력 등과 같은, 모바일 디바이스의 하드웨어 자원을 소비한다. 이렇게 소비된 자원은 사용자 애플리케이션 작업

에 필요한 자원과 경합할 수 있어, 사용자 애플리케이션 프로세스에 적용되는 자원이 너무 적은 경우, 사용자 인터페이스 글리치(glitches), 느리게 느껴지는 디바이스, 터치에 대한 느린 반응, 및/또는 디바이스 로크업을 야기할 수 있다.

[0004] 이러한 문제를 완화하기 위한 하나의 방식은 사용자 애플리케이션 프로세스보다 낮은 중요도로 실행하는 데몬 및/또는 다른 백그라운드(background) 프로세스들을 갖는 것이다. 사용자 애플리케이션 프로세스보다 낮은 중요도로 실행하는 이러한 다른 프로세스들을 가짐으로써, 사용자 애플리케이션 프로세스는 더 많은 모바일 디바이스 자원을 소비할 수 있고, 모바일 디바이스는 더 큰 반응성을 갖는 것으로 사용자에게 보일 것이다. 이러한 시나리오가 갖는 문제는, 사용자 애플리케이션 프로세스가 데몬이나 다른 백그라운드 프로세스들 중 하나에 서비스 또는 정보를 요청하면, 데몬 또는 백그라운드 프로세스는 더 낮은 중요도로 요청을 서비스할 것이라는 것이다. 이로 인해, 사용자 애플리케이션 프로세스는 서비스 요청을 실현하기 위해 대기하게 될 수 있다.

발명의 내용

해결하려는 과제

[0005] 데몬 프로세스의 중요도를 변경하는 디바이스의 방법 및 장치에 대해 설명한다. 예시적인 실시예에서, 디바이스는 데몬 프로세스로 향하는 사용자 프로세스로부터의 메시지를 수신하는데, 데몬 프로세스는 사용자 프로세스와 독립적으로 실행하고, 제1 데몬 프로세스는 다른 실행 프로세스들과 메시지 통신을 한다. 디바이스는, 제1 메시지가 제1 데몬 프로세스의 중요도가 변경될 수 있음을 나타내는지를 더 결정한다. 제1 메시지가, 제1 데몬 프로세스의 중요도가 변경될 수 있음을 나타내는 경우, 디바이스는 제1 데몬 프로세스의 중요도를 변경한다. 디바이스는 추가로 제1 데몬 프로세스에게 제1 메시지를 포워드한다.

[0006] 또 다른 실시예에서, 디바이스는 적응형 데몬으로 향하는 중요도 제공자 프로세스(importance donor process)로부터의 메시지를 수신하는데, 적응형 데몬은 아직 론치되지 않았다. 디바이스는 메시지가 적응형 데몬 프로세스의 중요도가 변경될 수 있음을 나타내는지를 더 결정한다. 메시지가 적응형 데몬 프로세스의 중요도가 변경될 수 있음을 나타내는 경우, 디바이스는 적응형 데몬 프로세스와 관련된 디폴트 중요도와는 다른 중요도를 갖는 적응형 데몬 프로세스를 론치한다. 추가로, 론치된 적응형 데몬 프로세스에 메시지를 포워드한다.

[0007] 다른 방법 및 장치에 대해서도 설명한다.

도면의 간단한 설명

[0008] 본 발명은, 비슷한 참조 번호가 유사한 요소를 나타내는 첨부 도면에서 제한으로서가 아닌 예로서 도시된다.

도 1은 낮은 중요도로 실행하는 데몬 프로세스들과 통신하는 사용자 애플리케이션 프로세스를 갖는 모바일 디바이스의 하나의 실시예의 블록도이다.

도 2는 낮은 중요도와 높은 중요도의 혼합으로 실행하는 데몬 프로세스들과 통신하는 사용자 애플리케이션 프로세스를 갖는 모바일 디바이스의 하나의 실시예의 블록도이다.

도 3a-b는, 데몬 프로세스의 중요도가 프로모트된, 데몬 프로세스와 통신하는 사용자 애플리케이션 프로세스를 갖는 모바일 디바이스의 하나의 실시예의 블록도이다.

도 4는 데몬으로 향하는 메시지에 기초하여 데몬 중요도를 프로모트하는데 사용되는 메시지를 전송하는 커널의 프로세스의 하나의 실시예의 흐름도이다.

도 5는 데몬으로 향하는 메시지에 기초하여 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 커널의 프로세스의 하나의 실시예의 흐름도이다.

도 6은 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 적응형 데몬의 프로세스의 하나의 실시예의 흐름도이다.

도 7은 중요도 어써션(importance assertion)을 드롭하는 커널의 프로세스의 하나의 실시예의 흐름도이다.

도 8은 요구에 따라(on demand) 서비스를 론치하는 프로세스의 하나의 실시예의 흐름도이다.

도 9는 적응형 데몬 프로세스 중요도를 프로모트(promote) 및 디모트(demote)하는 시스템의 하나의 실시예의 블록도이다.

도 10은 제1 데몬에 의해 수신된 메시지에 기초하여 제2 데몬 중요도를 프로모트하는 프로세스의 하나의 실시예의 흐름도이다.

도 11은, 제2 데몬 프로세스의 중요도가 프로모트된, 제2 데몬 프로세스와 통신하는 제1 데몬 프로세스를 갖는 모바일 디바이스의 하나의 실시예의 블록도이다.

도 12는 프로세스들에 의한 메시지의 전송 및 수신을 용이하게 하는 메시지 서비스 모듈이다.

도 13은 데몬에 의해 수신된 메시지에 기초하여 데몬 중요도를 프로모트하는데 사용되는 메시지를 전송하는 메시지 전송 모듈이다.

도 14는 데몬의 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 메시지 수신 모듈이다.

도 15는 데몬에 의해 수신된 메시지에 기초하여 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 사용자 수신 모듈이다.

도 16은 중요도 어썬션을 드롭하는 중요도 어썬션 드롭 모듈이다.

도 17은 요구에 따라 서비스를 론치하는 서비스 론치 모듈이다.

도 18은 제1 데몬에 의해 수신된 메시지에 기초하여 제2 데몬 중요도를 프로모트하는 메시지 처리 모듈이다.

도 19는 여기에 설명된 실시예들과 함께 사용될 수 있는 일반적인 컴퓨터 시스템의 하나의 예를 도시한다.

도 20은 본 발명의 하나의 실시예에 사용될 수 있는 데이터 처리 시스템의 예를 보여준다.

발명을 실시하기 위한 구체적인 내용

- [0009] 데몬 프로세스의 중요도를 변경하는 디바이스의 방법 및 장치에 대해 설명한다. 다음 설명에서는, 본 발명의 실시예에 대한 철저한 설명을 제공하기 위해 수많은 특정 상세가 명시되어 있다. 그러나, 본 발명의 실시예들은 이러한 특정 상세 없이 구현될 수 있다는 것이 당업자에게 명백할 것이다. 다른 경우에, 설명의 이해를 모호하게 하지 않기 위해, 잘-알려진 컴포넌트들, 구조들, 및 기술들은 자세히 제시되지 않았다.
- [0010] 명세서에서, "하나의 실시예" 또는 "일 실시예"라는 지칭은, 실시예와 관련하여 설명된 특정한 특징, 구조 또는 특성이 본 발명의 적어도 하나의 실시예에 포함될 수 있다는 것을 의미한다. 명세서의 다양한 장소에 등장하는 "하나의 실시예에서"라는 문구가 반드시 모두 동일한 실시예를 지칭하는 것일 필요는 없다.
- [0011] 다음의 설명 및 청구항들에서, "결합된(coupled)" 및 "접속된(connected)"이라는 용어가 그들의 파생어와 함께 사용될 수 있다. 이러한 용어들이 서로에 대한 동의어로서 의도된 것이 아님을 이해해야 한다. "결합된"은 두 개 이상의 요소가 서로 협력하거나 상호작용한다는 것을 표시하는데 사용되는데, 이들은 서로 직접적 물리적 또는 전기적 접촉 중일수도 있고 아닐 수도 있다. "접속된"은 서로 결합되는 두 개 이상의 요소 사이의 통신의 수립을 표시하는데 사용된다.
- [0012] 다음의 도면에 도시된 프로세스들은, (예컨대, 회로, 전용 논리 등의) 하드웨어, (범용 컴퓨터 시스템 또는 전용 기계에서 실행되는 것과 같은) 소프트웨어, 또는 이 둘의 조합을 포함하는 처리 논리에 의해 수행된다. 프로세스는 어떤 순차적인 동작들의 측면에서 아래에 설명되어 있지만, 설명된 동작들 중 일부는 다른 순서로 수행될 수 있다는 것을 이해해야 한다. 또한, 일부 동작들은 순차적이기 보다는 병렬로 수행될 수 있다.
- [0013] "서버", "클라이언트", 및 "디바이스"라는 용어는, 서버, 클라이언트 및/또는 디바이스에 대한 특정한 폼 팩터를 구체적으로 지칭하기 보다는 데이터 처리 시스템을 일반적으로 지칭하도록 의도된다.
- [0014] 데몬 프로세스의 중요도를 변경하는 디바이스의 방법 및 장치에 대해 설명한다. 하나의 실시예에서, 디바이스는, 중요도가 더 높은 프로세스로부터 적응형 데몬 프로세스로 전송되는 메시지를 검출하는 디바이스의 커널을 가짐으로써, 데몬 프로세스의 중요도를 변경할 수 있다. 적응형 데몬 프로세스는, 적응형 데몬 프로세스가 실행 중인 환경에 따라 중요도가 높아지거나 낮아질 수 있는 데몬 프로세스이다. 이러한 실시예에서, 커널은 메시지를 검출하고, 메시지를 중요도로 마크한다. 또한, 이러한 데몬 프로세스가 실행 중인 경우, 적응형 프로세스가 커널로부터 메시지를 수신하기 이전에, 커널은, 적응형 데몬 프로세스의 중요도를 프로모트할 수 있거나, 중요도가 더 높은 적응형 데몬을 론치할 수 있다. 적응형 데몬 프로세스가 메시지를 수신하기 이전에, 적응형 데몬 프로세스의 중요도가 프로모트됨으로써, 적응형 데몬 프로세스는 중요도가 더 높은 프로세스로서 메시지를 수신 및 처리할 수 있다. 또한, 커널은 적응형 데몬 프로세스에 대한 어썬션을 취한다.

- [0015] 적응형 데몬 프로세스가 메시지를 수신한 후에, 적응형 데몬 프로세스는 메시지를 처리하고 초기 메시지의 발신자에게 메시지에 대한 응답을 전송한다. 또한, 적응형 데몬 프로세스는 적응형 데몬 프로세스에 대한 어썬션을 드롭한다. 커널은 적응형 데몬 프로세스에 대한 보류중인 어썬션들의 수를 감소시킨다. 적응형 데몬 프로세스에 대해 보류중인 어썬션의 수가 0인 경우, 커널은 적응형 데몬 프로세스의 중요도를 그 적응형 데몬 프로세스의 디폴트 중요도로 디모트한다. 하나의 실시예에서, 데몬의 디폴트 중요도는, 프로모트되지 않은 경우 또는 프로모트된 중요도로 론치되지 않은 경우에 데몬이 갖는 중요도이다.
- [0016] 도 1은 낮은 중요도로 실행하는 데몬 프로세스와 통신하는 사용자 애플리케이션 프로세스를 갖는 모바일 디바이스(100)의 하나의 실시예의 블록도이다. 하나의 실시예에서, 모바일 디바이스(100)는 스마트폰, 랩톱, PDA, 음악 재생 디바이스, 게임 디바이스 등일 수 있다. 여기서 디바이스는 모바일 디바이스로서 설명되지만, 여기에 설명된 본 발명은 또한 개인용 컴퓨터, 서버 또는 여러 프로세스를 실행하는 임의의 다른 디바이스에서 사용될 수 있다. 도 1에서, 모바일 디바이스(100)는, 컴퓨터 하드웨어 자원들을 관리하는데 사용된 소프트웨어의 세트이고, 애플리케이션 프로그램들과 같은 실행중인 다른 컴퓨터 프로그램들을 위한 공통 서비스들을 제공하는 운영 체제(102)를 포함한다. 하나의 실시예에서, 운영 체제(102)는, 모바일 디바이스(100)의 프로세서가 실행중인 각 프로세스를 실행할 때 타임 스케줄링에 의해 실행중인 상이한 프로세스들을 관리한다. 하나의 실시예에서, 프로세스는 실행되고 있는 컴퓨터 프로그램의 인스턴스이다. 이러한 실시예에서, 프로세스는 사용자 입력의 결과로 실행하는 사용자 애플리케이션일 수 있다. 프로세스의 또 다른 예는 사용자 애플리케이션, 또 다른 데몬, 다른 프로세스 등에 하나 이상의 서비스를 제공하는 데몬 프로세스이다. 예를 들어, 하나의 실시예에서, 데몬 프로세스는 무선 핫스팟 서비스의 상태와 설치된 애플리케이션들의 목록을 제공하고, 검색을 용이하게 하고, 디바이스 전력 세팅을 모니터링하고 조정하는 등을 한다. 하나의 실시예에서, 데몬 프로세스는 모바일 디바이스의 특정한 하드웨어 자원에 결속되지(tied to) 않는다.
- [0017] 하나의 실시예에서, 일부 프로세스는 다른 프로세스보다 중요도가 높거나 낮을 수 있다. 하나의 실시예에서, 프로세스 중요도는, 디바이스와 현재 상호작용하는 사용자를 대신하여 액션이 필요하다는 것을 나타내는 작업/프로세스와 관련된 속성들의 수집이다. 그것은, 사용자가 이러한 액션이 완료되기를 기다리고 있고, 이러한 작업/프로세스는 긴급하게 처리될 것임을 나타낸다. 프로세스 중요도는 프로세스를 위한 중앙 처리 유닛 스케줄 우선 순위, 입/출력 스케줄링, 네트워크 대역폭 우선 순위, 메모리 관리, 그래픽 자원 사용 등 및/또는 이들의 조합을 포함할 수 있다. 프로세스 우선 순위는, 실행되기 위해 많은 작업이 필요할 때, 실행(run 또는 execute)을 위해 그 작업 또는 스레드가 긴급하게 선택된다는 것을 나타내는 속성이다. 이러한 프로세스의 우선 순위는 프로세서가 실행될 때를 스케줄링하기 위해 운영 체제에 의해 사용된다. 하나의 실시예에서, 다른 프로세스보다 중요도가 더 높은 프로세스는 상당한 양의 모바일 디바이스 자원(예컨대, 프로세서 자원 등)을 소비한다. 이러한 실시예에서, 중요도가 높은 프로세스는 중요도가 낮은 프로세스보다 더 빨리 실행하는 것으로 나타난다. 하나의 실시예에서, 중요도는 낮은 중요도에서 높은 중요도로 어느 한 범위의 숫자에 기초하여 할당될 수 있다. 예를 들어, 하나의 실시예에서, 중요도 범위는 1-10, 0-127, 낮은/높은, 또는 또 다른 범위의 값들에 이르는 것일 수 있다.
- [0018] 예를 들면, 하나의 실시예에서, 다음과 같은 모바일 디바이스 자원 사용은 프로세스 중요도에 의해 영향을 받을 수 있다. 1. 프로세서 스케줄링(Processor Scheduling). 이상에서 설명한 바와 같이, 중요도가 더 높은 및 더 낮은 프로세스에는, 중요도가 더 낮은 프로세스가 중요도가 더 높은 프로세스를 방해하지 않는 방식으로 상이한 프로세서 스케줄링 "우선 순위"가 할당될 수 있다.
- [0019] 2. 디스크 스케줄링(Disk Scheduling). 하나의 실시예에서, 시스템 저장 유닛(예컨대, 마그네틱 하드 드라이브 및 솔리드 스테이트 하드 디스크 유닛)에 액세스할 때, 중요도가 더 높은 프로세스는 중요도가 더 낮은 프로세스에 비해 우선 순위가 부여된다. 또한, 중요도가 더 낮은 프로세스(즉, 백그라운드 상태에 있는 프로세스와 관련된 스레드)는 시스템 저장 유닛에 대한 그들의 액세스에 있어서 속도-제한될 수 있다.
- [0020] 3. 진입 네트워크 활동(Incoming Network Activity). 하나의 실시예에서, 진입 네트워크 트래픽은 중요도가 더 낮은 프로세스로 향하는 일부 또는 모든 패킷을 드롭하여 완화될 수 있다. 또한, 부자연스럽게(artificially) 작은 버퍼 크기는 진입 TCP(전송 제어 프로토콜) 트래픽을 스로트하도록 말단의 네트워크 사이트(distal network sites)에 보고될 수 있다. 당업자는, TCP가 버퍼 크기에 관한 기존의 트래픽 흐름 제어 메커니즘 세트를 가지므로, 트래픽은 수신자가 처리할 수 있는 것보다 더 빠른 네트워크를 통해 전송되지 않는다는 것을 인식할 것이다. 전술한 바와 같이, 하나의 실시예에서, 부자연스럽게 작은 버퍼 크기가 보고될 수 있어, 전송자는 진입 트래픽의 흐름을 느리게 하거나 중지한다.

- [0021] 4. 진출 네트워크 활동(Outgoing Network Activity). 하나의 실시예에서, 네트워크 액세스는 두 개(2)의 큐, 즉, 중요도가 더 높은 프로세스와 관련된 작업을 위한 것 및 중요도가 더 낮은 프로세스와 관련된 작업을 위한 것의 사용을 통해 증대된다. 중요도가 더 낮은 큐보다 중요도가 더 높은 큐에 우선 순위를 부여하는 것이 유익하다는 것이 발견되었다. 예로서, 중요도가 더 높은 큐에게는 백그라운드 큐보다 100% 우선 순위가 부여될 수 있다. 이러한 세팅에서, 중요도가 더 낮은 큐에 있는 작업이 서비스되는 유일한 시간은 중요도가 더 높은 큐가 비어있는 경우이다. 또 다른 실시예에서, 중요도가 더 높은 큐에 있는 작업은 중요도가 더 낮은 큐로부터의 작업들과 임의의 원하는 비율, 예컨대, 90/10, 80/20, 75/25 또는 50/50으로 서비스될 수 있다. 또 다른 실시예에서, 일단 작업이 큐(예컨대, 중요도가 더 높은 큐)에 배치되면, 작업과 관련된 프로세스가 뒤이어 상이한 동작 상태(예컨대, 더 낮은 중요도)로 배치되는 경우에도 작업은 이동되지 않는다. 또 다른 실시예에서, 작업의 큐 위치는 그의 관련된 프로세스의 동작 상태를 반영하도록 업데이트된다.
- [0022] 5. 하드웨어 제약(Hardware Restrictions). 하나의 실시예에서, 중요도가 낮은 프로세스는 특정 시스템의 하드웨어 자원에 액세스하지 못하도록 방지된다. 중요도가 낮은 프로세스에 사용할 수 없는 예시적인 하드웨어는, 카메라, GPU, 가속도계, 자이로스코프, 메모리 소비량, 근접 센서 및 마이크를 포함하지만, 이에 한정되지 않는다.
- [0023] 하나의 실시예에서, 사용자 애플리케이션 프로세스(104)는 포어그라운드에서 실행할 수 있는 한편, 하나 이상의 데몬 프로세스(106A-N)는 사용자 애플리케이션 프로세스(104)보다 낮은 중요도로 백그라운드에서 실행한다. 포어그라운드에서 실행하는 프로세스는 사용자와 상호작용하는 프로세스이고, 사용자와 상호작용하지 않고 실행중인 프로세스로는 백그라운드에서 실행 중인 프로세스들이 있다. 예를 들어, 하나의 실시예에서, 모바일 디바이스(100) 상의 포어그라운드에서 실행 중인 사용자 애플리케이션 프로세스(104)는 모바일 디바이스(100)에 표시되는 사용자 애플리케이션일 수 있다. 이러한 실시예에서, 모바일 디바이스(100)는 모바일 디바이스에 표시되지 않고 대신 다른 프로세스들과 통신 중인 하나 이상의 데몬 프로세스를 동시에 실행할 수 있다. 이러한 데몬 프로세스는 백그라운드에서 실행 중인 것으로 사용자에게 직접적으로 보이지 않을 것이다.
- [0024] 하나의 실시예에서, 포어그라운드 및 중요도가 낮은 프로세스는 서로 다른 우선 순위로 실행할 수 있다. 포어그라운드 프로세스는 사용자와 상호작용하기 때문에, 포어그라운드 프로세스는 중요도가 낮은 프로세스인 데몬 프로세스보다 더 높은 중요도로 실행된다. 포어그라운드 프로세스는 더 높은 중요도로 실행함으로써, 포어그라운드 프로세스가 더 빨리 실행함에 따라, 포어그라운드 프로세스는 더 많은 프로세서 자원들을 얻어 포어그라운드 프로세스에게 더 나은 사용자 경험을 제공할 것이다.
- [0025] 하나의 실시예에서, 운영 체제(102)는 사용자 애플리케이션 프로세스(104) 및 데몬 프로세스(106A-N)를 포함한다. 본 실시예에서, 사용자 애플리케이션 프로세스(104)는 중요도가 더 높은 프로세스(108)이고, 데몬 프로세스(106A-N)는 중요도가 더 낮은 프로세스(110)이다. 또한, 사용자 애플리케이션 프로세스(104)는 더 낮은 중요도로 실행 중인 개개의 데몬 프로세스(106A-N)에 비해 모바일 디바이스(100) 자원에 대한 더 높은 점유율(share)을 얻을 것이다. 사용자 애플리케이션 프로세스(104)가 더 높은 중요도로 실행함으로써, 사용자 애플리케이션 프로세스(104)는 더 빨리 실행하는 것으로 사용자에게 나타날 것이다.
- [0026] 위에서 논의한 바와 같이, 이러한 배열의 한 가지 문제는, 사용자 애플리케이션 프로세스(104)가 가끔은 데몬 프로세스(106A-B)로부터 하나 이상의 서비스를 요청할 수 있다는 것이다. 본 실시예에서는, 데몬 프로세스(106A-B)가 더 낮은 중요도로 실행하기 때문에, 데몬이 이러한 요청을 서비스하는데 더 오래 걸릴 수 있다. 더 낮은 중요도로 실행 중인 하나 이상의 데몬 프로세스(106A-B)로부터 서비스를 요청하는 사용자 애플리케이션 프로세스(104)에 의해, 사용자 애플리케이션 프로세스(104)는, 데몬(106A-B)이 요청을 서비스하는 시간 동안 더 낮은 중요도로 실행하는 것으로 나타난다. 예를 들어, 하나의 실시예에서, 사용자 애플리케이션 프로세스(104)는 데몬으로부터 모바일 핫스팟의 상태를 요청하고, 전력 관리를 조정 및 보고하고, 검색을 수행하는 등을 한다.
- [0027] 도 2는 낮은 중요도와 높은 중요도의 혼합으로 실행하는 데몬 프로세스들과 통신하는 사용자 애플리케이션 프로세스를 갖는 모바일 디바이스(200)의 하나의 실시예의 블록도이다. 도 2에서, 하나의 실시예에서, 모바일 디바이스(200)의 운영 체제(202)는 중요도가 높은 프로세스(208) 중 하나로서 실행하는 데몬 프로세스(212)를 갖는다. 또한, 운영 체제(202)는 중요도가 낮은 프로세스(210)로서 실행하는 다른 데몬 프로세스(206A-N) 및 사용자 애플리케이션 프로세스(204)를 포함한다. 또한, 사용자 애플리케이션 프로세스(204)는 중요도가 낮은 프로세스인 데몬(206A-B)과 중요도가 높은 프로세스인 데몬(212)에 서비스 요청을 한다.
- [0028] 데몬 프로세스(212)가 중요도가 높은 프로세스(208) 중 하나로서 실행함으로써, 사용자 애플리케이션 프로세스

(204)는 데몬 프로세스(212)에 서비스 요청함으로써 성능에 있어서의 드롭을 경험하지 않는다. 이러한 데몬 프로세스(212)의 경우, 데몬 서비스 요청은 사용자 애플리케이션 프로세스(204)에 대한 성능 드롭을 완화한다. 그러나, 이러한 데몬(212)은 또한 운영 체제(202)에서 실행중인 다른 데몬 또는 다른 프로세스로부터의 서비스 요청을 처리할 수 있기 때문에, 또 다른 문제가 발생한다. 다른 서비스 요청들을 처리함으로써, 중요도가 높은 데몬 프로세스(212)는 많은 모바일 디바이스(200) 자원을 수신하여, 사용자 애플리케이션 프로세스(204)에 사용 가능한 자원의 양을 줄인다. 이러한 문제는 중요도가 높은 프로세스(208)로서 실행하는 데몬 프로세스의 수가 증가함에 따라 악화될 수 있다. 또한, 이러한 상황은, 중요도가 낮은 데몬이 낮은 중요도로 서비스 요청을 처리하게 되는 문제에도 역시 시달린다.

[0029] 도 1과 2는, 중요도가 낮은 프로세스로서 실행하거나 낮은 우선 순위 프로세스 및 높은 우선 순위 프로세스의 혼합으로서 실행하는 데몬 프로세스 모두에 서비스 요청을 함으로써, 사용자 애플리케이션 프로세스의 성능이 저하되는 문제를 예시한다. 하나의 실시예에서, 사용자 애플리케이션 프로세스 서비스 요청에 대해 높은 중요도로 프로모트되고 다른 서비스 요청에 대해 낮은 중요도로 실행되는 데몬 프로세스 중요도를 가짐으로써, 사용자 애플리케이션 프로세스가 더 나은 성능을 가질 수 있도록 한다. 이는, 사용자 애플리케이션 프로세스가 데몬 프로세스에 서비스 요청을 할 때, 이 서비스 요청은 높은 중요도로 처리되기 때문이다. 서비스 요청이 더 높은 프로세스 중요도로 처리됨으로써, 사용자 애플리케이션 프로세스는 더 빠르게 실행하는 것으로 나타날 것이다. 다른 서비스 요청에 대해, 데몬은 더 낮은 중요도로 실행되어, 다른 기능에 대한 사용자 애플리케이션 프로세스의 성능을 저하시키지 않는다.

[0030] 하나의 실시예에서, 프로세스는 하나 이상의 스레드를 포함할 수 있고, 여기서, 스레드는 운영 체제에 의해 스케줄링될 수 있는 최소 처리 단위(smallest unit processing)이다. 다수의 스레드가 동일한 프로세스 내에 존재할 수 있고, 프로세스 자원들(예컨대, 메모리, 명령어, 컨텍스트 등)을 공유할 수 있다. 하나의 실시예에서, 프로세스의 중요도가 변경되는 경우, 모든 스레드는 변경된 중요도로 실행될 것이다. 본 실시예에서, 메시지의 수신 또는 메시지 응답의 전송에 응답하여 중요도가 높아지거나 낮아질 때, 프로세스의 현재 실행중인 모든 스레드는 새로운 중요도로 실행될 것이다. 또 다른 실시예에서, 스레드 중 일부가 변경된 중요도로 실행된다.

[0031] 도 3a는, 데몬 프로세스(306A)의 중요도가 프로모트된, 데몬 프로세스(306A)와 통신하는 사용자 애플리케이션 프로세스(304)를 갖는 모바일 디바이스(300)의 하나의 실시예의 블록도이다. 도 3a에서, 모바일 디바이스(300)는 사용자 애플리케이션 프로세스(304) 및 데몬 프로세스(306A-N)의 실행을 관리하는 운영 체제(302)를 포함한다. 또한, 사용자 애플리케이션 프로세스(304)는 높은 중요도로 실행하고 데몬 프로세스는 처음에 낮은 중요도로 실행한다. 사용자 애플리케이션 프로세스(304)가 데몬 프로세스로부터 서비스를 요청할 때, 데몬 프로세스는 더 높은 중요도로 프로모트되어 이 서비스 요청을 처리한다. 하나의 실시예에서, 이 프로세스의 중요도는 프로세스가 동작하는 환경에 적응될 수 있기 때문에, 데몬 프로세스(306A)는 적응형 데몬이다. 예를 들어, 하나의 실시예에서, 사용자 애플리케이션 프로세스(304)는 메시지(312)를 통해 데몬 프로세스(306A)에게 서비스 요청을 한다. 하나의 실시예에서, 메시지는 프로세스간 통신(IPC) 메시지이다. 데몬을 위해 큐잉되는 메시지에 응답하여, 데몬의 중요도는 더 높은 중요도의 데몬(316)으로 프로모트된다. 하나의 실시예에서, 프로모트된 데몬(316)의 중요도는 요청하는 사용자 애플리케이션 프로세스(304)와 동일하지만, 대안적인 실시예에서, 프로모트된 데몬(316)의 중요도는 요청하는 사용자 애플리케이션 프로세스(304)의 중요도보다 더 높거나 더 낮을 수 있다.

[0032] 또한, 사용자 애플리케이션 프로세스(304)는 데몬 프로세스(306A)에 메시지를 전송한 후 및 데몬 프로세스(306A)로부터 해당 메시지에 대한 응답을 수신하기 전에 계속 실행된다. 예를 들어, 하나의 실시예에서, 사용자 애플리케이션 프로세스(304)가 데몬 프로세스(306A)에 메시지를 전송한 후 및 사용자 애플리케이션 프로세스(304)가 해당 메시지에 대한 응답을 수신하기 전에, 사용자 애플리케이션 프로세스(304)는 사용자로부터 입력을 수신할 수 있고, 사용자 애플리케이션 프로세스(304)에 대한 디스플레이를 업데이트할 수 있고(예컨대, 스크롤, 크기조정, 회전 등), 및/또는 다른 기능을 수행할 수 있다. 또 다른 예로서, 사용자 애플리케이션 프로세스(304)는 데몬 프로세스(306A/316)로부터 응답을 기다리는 동안 중단되지 않는다.

[0033] 하나의 실시예에서, 데몬 프로세스(306A)는 사용자 애플리케이션 프로세스(304)가 데몬 프로세스(306A)로 서비스 요청 메시지를 전송하는 시간에 실행한다. 본 실시예에서, 데몬 프로세스(306A)는 사용자 애플리케이션 프로세스(304)와 독립적으로 실행하고, 사용자 애플리케이션 프로세스(304), 다른 사용자 애플리케이션 프로세스(들), 다른 데몬(들), 다른 프로세스 및/또는 이들의 조합으로부터의 서비스 요청을 처리한다. 또한, 데몬 프로세스(306A)는 사용자 애플리케이션 프로세스(304)가 시작되기 전 또는 후에 시작되었을 수 있다. 예를 들어, 하나의 실시예에서, 모바일 디바이스가 부팅될 때 데몬 프로세스(306A)는 시작되고, 모바일 디바이스가 부팅된

후에 사용자 애플리케이션 프로세스(304)가 시작된다. 본 예에서, 데몬 프로세스(306A)는 사용자 애플리케이션 프로세스(304)와 독립적으로 모바일 디바이스에서 실행중인 하나 이상의 프로세스로부터의 서비스 요청 메시지를 처리한다.

[0034] 또 다른 실시예에서, 데몬 프로세스(306A)는 사용자 애플리케이션 프로세스(304)가 메시지를 전송할 때 실행되지 않는다. 본 실시예에서, 데몬 프로세스(306A)는 일반적으로 갖는 것보다 높은 중요도로 요구에 따라 론치된다. 요구에 따른 론치는 아래의 도 8에서 더 설명되어 있다. 하나의 실시예에서, 론치 서비스(326)는 요구에 따라 데몬의 론칭(328)을 처리한다. 본 실시예에서, 서비스 요청 메시지가 론치 서비스(326)로 포워드되고, 론치 서비스(326)는, 보통은 더 낮은 중요도의 프로세스(306A)로서 론치하는 데몬(306A/316)을 더 높은 중요도의 프로세스(316)로서 론치한다.

[0035] 도 3a에서, 하나의 실시예에서, 운영 체제 커널(320)은 데몬 프로세스 중요도 프로모션을 처리한다. 또 다른 실시예에서, 운영 체제 커널(320)은 프로세스 관리 모듈(322)을 사용하여 데몬 프로세스 중요도 프로모션을 처리한다. 본 실시예에서, 프로세스 관리 모듈(322)은 사용자 애플리케이션 프로세스(304)와 데몬 프로세스(306A/316) 사이에서 메시지의 전송 및 수신을 처리한다. 예를 들어, 하나의 실시예에서, 프로세스 관리 모듈(322)은, 프로세스 관리 모듈(322)이 데몬 프로세스(306A/316)에 대한 메시지를 인큐할 때 데몬 프로세스(306A/316) 프로모션을 프로모트한다. 이러한 두 가지 프로세스 사이에서 메시지 전달을 처리함으로써, 프로세스 관리 모듈(322)은 데몬 프로세스 프로모션을 관리할 수 있다. 또한, 데몬 프로세스(306A/316)는 수신된 메시지를 처리하는 데몬 msg_recv 모듈(324)을 포함한다. 수신된 메시지를 처리하는 것은 아래의 도 6에 더 설명되어 있다.

[0036] 하나의 실시예에서, 데몬이 메시지의 처리를 완료한 후에, 데몬 중요도는 초기 데몬 중요도로 리턴된다. 도 3b는, 데몬 프로세스(316)의 중요도가 디모트된, 프로모트된 데몬 프로세스(316)와 통신하는 사용자 애플리케이션 프로세스(304)를 갖는 모바일 디바이스(300)의 하나의 실시예의 블록도이다. 도 3b에서, 모바일 디바이스(300)는, 사용자 애플리케이션 프로세스(304), 데몬 프로세스(306B-N), 및 프로모트된 데몬 프로세스(316)의 실행을 관리하는 운영 체제(302)를 포함한다. 도 3a에서와 같이, 사용자 애플리케이션 프로세스(304)는 높은 중요도로 실행되고, 데몬 프로세스(306B-N)는 초기에 낮은 중요도로 실행되며, 데몬(306A)은 더 높은 중요도로 실행하도록 프로모트되어, (프로모트된) 데몬 프로세스(316)로서 도시된다. 하나의 실시예에서, 프로모트된 데몬 프로세스(316)는 메시지를 처리하고 사용자 애플리케이션 프로세스(304)에 응답을 전송한다. 본 실시예에서, 메시지의 전송에 응답하여, 프로모트된 데몬 프로세스(316)는 초기 중요도(306A)로 다시 디모트될 수 있다. 또한, 데몬 프로세스(316) 디모션은, 프로모트된 중요도로 실행 또는 론치되는 동안 데몬 프로세스(306A) 중요도가 프로모트되는 경우에 발생할 수 있다. 프로세스 디모션은 아래에 더 설명되어 있는데, 여기서, 프로모트된 데몬 프로세스(316)는 어썬션을 드롭하여 이 프로세스(316)가 디모트되도록 이어질 수 있다. 예를 들어, 프로세스 디모션은 아래의 도 6에 더 설명되어 있다.

[0037] 도 3b에서, 하나의 실시예에서, 운영 체제 커널(320)은 데몬 프로세스 디모션을 처리한다. 또 다른 실시예에서, 운영 체제 커널(320)은 프로세스 관리 모듈(322)을 사용하여 데몬 프로세스 프로모션을 처리한다. 본 실시예에서, 프로세스 관리(322)는 사용자 애플리케이션 프로세스(304)와 데몬 프로세스(306A/316) 사이에서 메시지의 전송 및 수신을 처리한다. 이러한 두 가지 프로세스 사이에서 메시지 전달을 처리함으로써, 프로세스 관리 모듈(322)은 데몬 프로세스 디모션을 관리할 수 있다. 또한, 데몬 프로세스(306A/316)는 수신된 메시지를 처리하는 데몬 msg_recv 모듈(324)을 포함한다. 수신된 메시지의 처리는 아래의 도 6에 더 설명되어 있다.

[0038] 하나의 실시예에서, 프로세스 디모션은 프로모트된 데몬 프로세스(316)가 어썬션을 드롭함으로써 발생한다. 예를 들어, 하나의 실시예에서, 프로세스 어썬션 드롭은 커널에 시스템 호출을 하는 데몬 프로세스에 의해 발생한다. 이 시스템 호출은 데몬 프로세스가 사용자 애플리케이션 프로세스 대신 작업을 완료했음을 커널에게 나타낸다. 본 실시예에서, 어썬션은 메시지가 처리되었고 프로세스가 디모트될 수 있다는 것을 운영 체제에 통보한다. 하나의 실시예에서, 운영 체제는, 이 데몬의 프로모션에 대해 보류중인 어썬션의 수를 감소시킨다. 본 실시예에서, 데몬 프로세스의 중요도가 프로모트될 것임을 나타내는 다른 어썬션이 없는 경우, 프로모트된 데몬 프로세스(316)의 중요도는 디폴트 중요도로 디모트된다. 대안적으로, 이 데몬에 대해 다른 보류중인 어썬션이 존재하는 경우, 운영 체제는 이 데몬의 프로모션에 대해 보류중인 어썬션의 수를 감소시킨다. 하나의 실시예에서는, 데몬 프로세스의 중요도는 데몬 프로세스의 디폴트 중요도로 디모트되는 한편, 대안적인 실시예들에서는, 디모트된 프로세스 중요도는 디폴트 중요도보다 더 낮거나 더 높을 수 있다. 하나의 실시예에서, 디폴트 중요도는 중요도가 변경되지 않은 프로세스의 중요도이다.

- [0039] 예를 들면, 하나의 실시예에서, 선호 애플리케이션(preferences application)이 론치되고 사용자가 모바일 디바이스의 네트워크 세팅을 체크하려고 내비게이트할 때의 예를 고려한다. 이 예에서, 모바일 인터넷 공유 데몬(mobile Internet sharing daemon)은 데몬 프로세스(306A)로 표현될 것이며, 선호 애플리케이션은 사용자 애플리케이션 프로세스(304)일 것이다. 선호 애플리케이션은 개인용 핫스팟 서비스에 대한 상태 정보를 조회하기 위해 모바일 인터넷 공유 데몬(예컨대, 데몬 프로세스(306A))에 메시지를 전송한다. 모바일 인터넷 공유 데몬은, 메시지의 수신 시에, 캐리어에 대한 테더링이 사용가능한지 및 그 사용에 대한 권한이 사용자에게 부여되어 있는지를 결정한다. 모바일 인터넷 공유 데몬은 해당 정보를 선호 애플리케이션으로 다시 리턴하여, 개인용 핫스팟 서비스에 대한 적절한 상태를 표시한다.
- [0040] 하나의 실시예에서, 모바일 인터넷 공유 데몬은 선호 애플리케이션이 이 데몬에 메시지를 전송할 때 실행된다. 본 실시예에서, 이 데몬을 위한 서비스 포트는 모바일 인터넷 공유 데몬과 함께 존재할 것이다. 선호 애플리케이션으로부터 메시지가 전송될 때, 커널은, 포트로의 메시지의 큐잉의 일환으로, 메시지가 사용자 애플리케이션으로부터의 것이고 적응형 데몬으로 향한다는 것을 통지한다. 커널은 어썬션을 취하고, 이 데몬이 낮은 중요도(예컨대, 백그라운드 중요도)로 실행중인 경우 더 높은 중요도로 모바일 인터넷 공유 데몬을 프로모트한다. 또한, 메시지가 커널이 어썬션을 보유하게 한다는 것을 나타내도록 메시지가 마크된다. 모바일 인터넷 공유 데몬이 메시지를 수신할 때, 이 데몬은 특정한 메시지가 커널에 의한 어썬션으로 인한 것인지 여부를 체크하고, 만약 그렇다면, 처리 및 응답 이후에, 데몬은 커널에게 어썬션을 드롭하라고 지시한다. 어썬션 드롭시에, 보류중인 다른 어썬션이 없는 경우, 커널은 이 프로세스에 대한 디폴트 중요도(예컨대, 백그라운드 중요도)로 모바일 인터넷 공유 데몬을 다시 디모트할 것이다.
- [0041] 또 다른 실시예에서, 모바일 인터넷 공유 데몬이 실행중이지 않고, 이 데몬 프로세스는 데몬의 하나의 디폴트 중요도와는 상이한 중요도로 요구에 따라 론치된다. 예를 들어, 하나의 실시예에서, 데몬은 위에서 논의된 모바일 인터넷 공유 데몬인 것으로 간주한다. 모바일 인터넷 공유 데몬이 현재 실행중이지 않은 경우, 모바일 인터넷 공유 데몬에 대한 서비스 포트에는 론치 데몬이 저장되어 있다. 론치 데몬이, 포트가 메시지(예컨대, 이벤트 메시지)를 가지고 있다는 통지를 받을 때, 론치 데몬은, 적응형 데몬인, 모바일 인터넷 공유 데몬을 시작하는 프로세스 생성을 호출한다. 론치를 위한 인수(arguments) 중 하나로서, 론치 데몬은 커널에 문의중인 포트에 핸들을 제공한다. 운영 체제 커널은, 임의의 보류중인 메시지가 데몬 서비스 포트를 포함하는지 여부, 사용자 애플리케이션 프로세스로부터의 것인지 여부, 및, 또한, 데몬 프로세스를 프로모트하는 어썬션을 운반하는지 여부를 결정한다. 이러한 유형의 메시지가 존재하는 경우, 커널은 프로모트된 중요도로 모바일 인터넷 공유 데몬의 론치를 시작한다. 하나의 실시예에서, 프로모트된 중요도는 디폴트 모바일 인터넷 공유 데몬 중요도보다 높은 중요도일 수 있고, 사용자 애플리케이션 프로세스의 중요도와 동일하거나, 높거나, 낮은 것일 수 있다. 본 실시예에서, 커널은 이 시점에 프로세스에 대한 어썬션을 보유하고 있다. 데몬을 향한 메시지가 어썬션을 포함하지 않는 경우, 모바일 인터넷 공유 데몬은 프로세스에 대한 디폴트 중요도로 론치된다. 예를 들어, 하나의 실시예에서, 모바일 인터넷 공유 데몬은 백그라운드 중요도로 론치된다.
- [0042] 론치 모바일 인터넷 공유 데몬 프로세스가 일단 완료되면, 모바일 인터넷 공유 데몬은 론치 데몬과 랑데부(rendezvous)하고, 모바일 인터넷 공유 데몬으로 이전된 서비스 포트를 얻는다. 모바일 인터넷 공유 데몬은 메시지를 판독하고, 해당 메시지가 어썬션을 야기했는지를 체크하는데, 여기서 어썬션은 커널에 의해 마크된다. 메시지가 어썬션을 야기한 경우, 모바일 인터넷 공유 데몬이 메시지 처리를 완료하고 메시지에 응답할 때, 모바일 인터넷 공유 데몬은 어썬션을 드롭할 것이다. 보류중인 어썬션이 프로세스에서 드롭되면, 모바일 인터넷 공유 데몬은 디폴트 프로세스 중요도(예컨대, 백그라운드 중요도)로 돌아간다.
- [0043] 또 다른 예로서, 사용자가 모바일 디바이스를 사용하기 시작할 때, 애플리케이션 론치 애플리케이션(application launcher application)은 유틸 수면을 방지하기 위해, 전력 어썬션을 보유하는 전력 데몬으로 메시지를 전송한다. 하나의 실시예에서, 모바일 디바이스의 백라이트가 온(on)일 때까지, 전력 데몬은 이 어썬션을 보유한다. 본 실시예에서, 사용자가 디바이스를 잠고 어떠한 백그라운드 오디오도 재생되고 있지 않으면, 애플리케이션 론치 애플리케이션은 전력 어썬션을 드롭한다. 디바이스가 일정 시간 기간 동안 사용중이지 않은 경우, 모바일 디바이스 백라이트는 꺼진다. 애플리케이션 론치 애플리케이션은 전력 어썬션을 드롭하고, 커널이 디바이스의 유틸 수면을 가능하게 한다.
- [0044] 이러한 예에서, 전력 데몬은 적응형 데몬으로서, 이는, 전력 데몬이 애플리케이션 론치 애플리케이션과 같은 중요도가 더 높은 프로세스로부터의 메시지에 응답하여 프로모트될 수 있는 디폴트 중요도를 갖는다는 것을 의미한다. 이는 포어그라운드 애플리케이션 활동에 미치는 영향을 줄일 수 있다. 예를 들어, 하나의 실시예에서, 애플리케이션 론치 애플리케이션으로부터의 메시지가 전력 데몬에 도착했을 때, 전력 데몬은 더 높은 중요도로

부스트되고, 전력 데몬에 의한 메시지 처리의 완료시, 전력 데몬은 다시 전력 데몬의 디폴트 중요도(예컨대, 백그라운드 중요도)로 드롭될 것이다. 이러한 예는, 적응형 데몬이 메시지 처리 시간 동안 정상 중요도로 실행할 수 있으며 불필요하게 포어그라운드 애플리케이션에 영향을 미치지 않는 경우를 예시한다. 이러한 예는 또한 적응형 데몬이, 입력/출력, 처리 유닛 등과 같은 자원을 너무 많이 소비하거나 경합하지 않으면서 포어그라운드 애플리케이션의 반응성을 향상시키는데 도움이 되는지를 예시한다. 또한, 또 다른 데몬은, 전력 데몬의 중요도가 프로모트되지 않은 전력 데몬에 요청을 할 수 있다. 이러한 다른 데몬의 요청은 전력 데몬의 디폴트 중요도로 처리될 것이다.

[0045] 또 다른 예 및 실시예에서, 커널은 프로세스들 사이의 통신 채널(포트)의 수신단(receive end)의 소유권을 이전하는 능력을 갖는다. 예를 들어, 하나의 실시예에서, 이러한 이전은, 적응형 데몬이 론칭 데몬으로부터 서비스 포트의 소유권을 요청할 때 발생하는 것이다. 또 다른 예로서, 또 다른 실시예에, 적응형 데몬(A)은 포트(P)에 대한 수신 권리(receive right)를 갖는다. 이러한 예에서, 포트(P)에 인큐된 N 중요도 제공(donating) 메시지들이 존재하여, 데몬은 N 중요도 제공(donation)/부스트 레퍼런스를 갖는다. 적응형 데몬이 N 메시지를 수신하고, 그들을 처리한 다음, 각각 이후에 어썬션을 드롭하는 경우, 적응형 데몬은 정상적인 적응형 중요도로 다시 되돌아갈 것이다. 이러한 경우/인스턴스에서, 모든 N 메시지를 수신하는 대신, 적응형 데몬은, 상이한 포트(Q)로 향하는 메시지의 콘텐츠에서 포트(P)의 수신단을 전송함으로써, 해당 포트의 수신단의 소유권을 일부 또 다른 프로세스(B)로 이전한다. 이러한 경우에, 적응형 데몬(A)은 N 부스트들을 손실하고, 프로세스(B)(또한 적응형인 경우)는 수신-권리 이전 메시지가 인큐되는 시간에 N 부스트들을 획득한다.

[0046] 적응형 데몬(B)이, 포트(P)의 수신단을 포함하는 메시지를 수신하기 전에 포트(Q)의 수신단을 일부 다른 프로세스로 이전하는 경우, 적응형 데몬(B)은 포트(P)와 관련된 N 중요도 어썬션은 물론 포트 Q에서 다른 메시지에 대한 중요도 제공 어썬션(importance donation assertion)의 수를 이전해야 한다. 이러한 식으로, 중요도 제공 어카운팅(importance donation accounting)은 이들 포트 중 임의의 것에 인큐된 임의의 메시지들의 수신-권리 콘텐츠들에 따라 회귀적(recursive)이다.

[0047] 또 다른 예 및 실시예에서, 적응형 데몬(A)은 그것이 유휴 상태라고 결정하는 경우, 적응형 데몬(A)은 훨씬 더 많은 시스템 자원을 절약하기 위해 종료(exit)를 선택할 수 있다. 이 데몬 또는 저 데몬이 막 크래시(crash)되었다. 어느 경우든, 포트(P)에 대한 수신 권리는 론칭 데몬으로 다시 이전된다(예컨대, 운영 체제 프로세스간 통신 시스템에서 빌트-인 백업 메커니즘을 통해). 중요도-제공 메시지가 이러한 종료/크래시 직전에 인큐되는 경우, 적응형 데몬(A)은 실행을 중지하는 시간에 중요도 제공을 가질 것이다. 이러한 예에서, 커널은 적응형 데몬의 부스트가 영구적으로 손실되는 것을 원하지 않는다. 따라서, 론칭 데몬이 다시 포트(P)의 수신단을 얻을 때, 론칭 데몬은 인큐된 메시지가 이미 있다는 것을 검출하고, 적응형 데몬(A)의 다음 인스턴스(예컨대, A2로 불림)를 재론치(relaunch)한다. 적응형 데몬(A2)은 포트가 론칭 데몬으로 리턴된 후에 도착하는 중요도 제공 메시지들의 양은 물론, 이전에 이미 해당 포트의 인큐된 메시지에 의해 부스트될 필요가 있다. 이러한 "포트 마이그레이션(port migration)을 통한 중요도 업데이트" 메커니즘은 이것이 가능하도록 한다.

[0048] 도 4는 데몬에 의해 수신된 메시지에 기초하여 데몬 프로세스 중요도를 프로모트하는데 사용되는 메시지를 전송하는 커널의 프로세스(400)의 하나의 실시예의 흐름도이다. 프로세스는 하드웨어(예컨대, 회로, 전용 논리, 프로그래머블 논리, 마이크로코드 등), 소프트웨어(예컨대, 하드웨어 시뮬레이션을 수행하는 처리 디바이스에서 실행되는 명령어들) 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 하나의 실시예에서, 프로세스(400)는 도 3a 또는 3b의 커널(320)과 같은 커널에 의해 수행된다.

[0049] 도 4에서, 프로세스(400)는, 블록 402에서, 메시지를 전송하는 현재의 프로세스가 중요도 제공자인지를 결정하는 것으로 시작한다. 하나의 실시예에서, 중요도 제공자는, 사용자 애플리케이션 프로세스인 프로세스이고, 사용자는 액션이 완료하기를 기다린다. 또 다른 실시예에서, 중요도 제공자는 또한 사용자 애플리케이션 프로세스 대신 작업을 수행하는 데몬 또는 서비스이다. 본 실시예에서, 데몬은 프로모트된 중요도를 가질 수 있다. 하나의 실시예에서, 중요도 제공자는 포어그라운드에서 실행중인 사용자 애플리케이션 프로세스이다. 예를 들어, 하나의 실시예에서, 상기 도 3a에 설명된 바와 같이, 중요도 프로세스는 사용자 애플리케이션 프로세스(304)이다. 현재 프로세스가 중요도 제공자가 아닌 경우, 실행은 아래의 블록 414로 진행된다.

[0050] 현재 프로세스가 중요도 제공자인 경우, 블록 404에서, 프로세스(400)는, 메시지를 수신할 대상 프로세스가 적응형 데몬 프로세스인지를 결정한다. 대상 프로세스가 적응형 데몬 프로세스가 아닌 경우, 실행은 아래의 블록 412로 진행된다. 대상 프로세스가 중요도 수신자인 경우, 블록 406에서, 프로세스(400)는 중요도를 운반하는 것으로서 메시지를 마크하고, 대상 프로세스에 대한 "중요도 어썬션 어카운트"를 증가시킨다. 하나의 실시예에

서, "중요도 어썬션 카운트"를 증가시킴으로써, 프로세스(400)는 대상 프로세스에 대한 어썬션을 취한다. 이러한 어썬션 카운트는 어썬션의 중요도를 디모트할지 여부를 결정하기 위해 커널에 의해 나중에 사용된다. 본 실시예에서, 어썬션은 대상 프로세스에 의해 메시지를 처리하는 것과 관련하여 나중에 드롭된다. 어썬션을 드롭하는 것은 아래의 도 6에 더 설명되어 있다. 예를 들어, 하나의 실시예에서, 사용자가 네트워킹 상태를 알기를 원할 때, 사용자 선호 애플리케이션 프로세스는 네트워크 핫스팟 서비스의 상태에 대해 모바일 인터넷 공유 데몬으로 메시지를 전송한다. 위의 설명에 따라, 선호 애플리케이션은 사용자 애플리케이션 프로세스이며, 모바일 인터넷 공유 데몬은 적응형 데몬인데, 여기서, 모바일 인터넷 공유 데몬으로의 메시지는 중요도를 운반하는 것으로서 마크되고, 모바일 인터넷 공유 데몬에 대한 중요도 어썬션 카운트는 증가된다.

[0051] 블록 408에서, 프로세스(400)는 대상 프로세스가 낮은 중요도를 갖는지를 결정한다. 하나의 실시예에서, 프로세스(400)는 대상 프로세스가 현재 프로세스보다 낮은 중요도를 갖는지를 결정한다. 예를 들어, 하나의 실시예에서, 프로세스(400)는 현재 및 대상 프로세스의 중요도를 비교한다.

[0052] 프로세스(400)는, 블록 410에서, 대상 프로세스를 프로모트된 중요도로 변경한다. 하나의 실시예에서, 프로세스(400)는 대상 프로세스를 현재 프로세스의 중요도로 변경한다. 본 실시예에서, 프로세스(400)는 포트 큐에 메시지를 인큐할 때 또는 인큐하기 전에 그리고 대상 프로세스가 메시지 처리를 위해 메시지를 디큐하기 전에 대상 프로세스 중요도를 프로모트한다. 대상 프로세스가 메시지를 디큐하기 전에 대상 프로세스를 프로모트함으로써, 대상 프로세스는 더 높은 중요도로 메시지를 디큐 및 처리할 수 있다. 예를 들어, 하나의 실시예에서, 현재 프로세스는 포어그라운드 중요도로 실행할 수 있고, 프로세스(400)는 대상 프로세스 중요도를 포어그라운드 중요도로 변경한다. 또 다른 실시예에서, 프로세스(400)는 현재 프로세스의 중요도와는 다른 값으로 대상 프로세스 중요도를 변경한다. 하나의 실시예에서, 프로세스(400)는 도 3a에서 전술한 바와 같이 대상 프로세스의 중요도를 변경한다. 실행은 아래의 블록 412로 진행된다.

[0053] 위에서 블록 402에서 결정된 현재 프로세스가 중요도 제공자가 아닌 경우, 프로세스는 블록 414에서 메시지 포트가 서비스 포트인지를 결정한다. 하나의 실시예에서, 현재 프로세스가 중요도 제공자가 아닌 경우, 현재의 프로세스는 요구에 따라 론치될 데몬에 대한 메시지를 처리하는 론칭 데몬이다. 본 실시예에서, 서비스 포트는 해당 서비스의 대상 프로세스에 메시지를 전달하는데 사용되는 포트이다. 메시지 포트가 서비스 포트인 경우, 블록 416에서, 프로세스(400)는 중요도를 운반하는 것으로서 메시지를 마크하고, 서비스 포트에 대한 "중요도 어썬션 어카운트"를 증가시킨다. 하나의 실시예에서, 서비스 포트에 대한 "중요도 어썬션 카운트"를 증가시킴으로써, 프로세스(400)는 포트가 적응형 데몬 또는 또 다른 중요도 제공자에 의해 요구에 따라 론치되었을 수 있다는 것을 마크한다. 실행은 아래의 블록 412로 진행된다. 메시지 포트가 서비스 포트가 아닌 경우, 아래에서 실행은 블록 412로 진행된다.

[0054] 블록 412에서, 프로세스(400)는 포트에 메시지를 큐한다. 하나의 실시예에서, 포트는 대상 프로세스를 위해 사용되는 포트이다. 하나의 실시예에서, 마하 프로세스간 통신(Mach interprocess communication)(IPC) 포트가 지원되는 한편, 다른 실시예에서는, 다른 유형의 포트(예컨대, 로컬 도메인 소켓, 파이프 등), 임의의 다른 유형의 IPC 등이 지원된다.

[0055] 도 4는 데몬 프로세스의 중요도를 프로모트하는데 사용될 수 있는 메시지를 전송하기 위해 커널에 의해 사용되는 프로세스를 설명한다. 도 5는 데몬으로 향하는 메시지에 기초하여 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 커널의 프로세스(500)의 하나의 실시예의 흐름도이다. 프로세스는 하드웨어(예컨대, 회로, 전용 논리, 프로그래머블 논리, 마이크로코드 등), 소프트웨어(예컨대, 하드웨어 시뮬레이션을 수행하기 위해 처리 디바이스에서 실행된 명령어들), 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 하나의 실시예에서, 프로세스(500)는, 도 3a 또는 3b의 커널(320)과 같은, 커널에 의해 수행된다.

[0056] 도 5에서, 프로세스(500)는, 블록 502에서, 현재 프로세스에 대한 큐잉된 메시지가 존재하는지를 결정하는 것으로 시작한다. 하나의 실시예에서, 프로세스(500)는, 메시지가 현재의 프로세스, 이를테면, 도 3a에서 전술한 바와 같은 데몬 프로세스와 연관된 포트에 큐되는지를 결정한다. 하나의 실시예에서, 현재 프로세스는 적응형 데몬이다. 큐된 메시지가 존재하지 않는 경우, 실행 프로세스(500)는 종료된다.

[0057] 메시지가 존재하는 경우, 프로세스(500)는, 블록 504에서, 포트로부터 메시지를 디큐한다. 하나의 실시예에서, 프로세스(500)는 포트 큐로부터 해당 메시지를 제거한다. 프로세스(500)는, 블록 506에서 메시지가 중요도에 대해 마크되었는지를 결정한다. 하나의 실시예에서, 프로세스(500)는, 메시지가 중요도에 대해 마크되었는지를 결정하기 위해 메시지를 검사한다. 예를 들어, 하나의 실시예에서, 메시지는 중요도 비트를 포함하고, 중요도 비트가 설정되어 있으면, 메시지는 중요도에 대해 마크되어 있다. 메시지가 중요도에 대해 마크되어 있지 않으면

면, 실행 프로세스는 아래의 블록 512로 진행된다.

- [0058] 메시지가 중요도에 대해 마크되어 있는 경우, 블록 508에서, 프로세스(500)는 현재 프로세스가 적응형 데몬 프로세스인지를 결정한다. 현재 프로세스가 적응형 데몬 프로세스가 아닌 경우, 블록 510에서, 프로세스(500)는 메시지가 중요도를 운반한다는 것을 표시하는 플래그를 제거한다. 하나의 실시예에서, 프로세스(500)는 메시지의 중요도 비트를 언셋한다. 또한, 블록 510에서, 프로세스(500)는 포트 중요도 카운트를 감소시킨다. 본 실시예에서, 포트 중요도 카운트는, 중요도 제공자 프로세스로부터의 대기중인(outstanding) 메시지의 수를 나타내는 카운터이다. 실행 프로세스는 아래의 블록 512로 진행된다. 현재 프로세스가 적응형 프로세스인 경우, 실행 프로세스는 아래의 블록 512로 진행된다.
- [0059] 블록 512에서, 프로세스(500)는 사용자 버퍼로 메시지를 복사한다. 하나의 실시예에서, 사용자 버퍼에 메시지를 복사함으로써, 현재 프로세스가 메시지를 처리할 수 있다. 하나의 실시예에서, 현재 프로세스는 도 6에 후술하는 바와 같이 메시지를 처리하도록 프로세스(600)를 수행한다.
- [0060] 도 6은 데몬에 의해 수신된 메시지에 기초하여 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 적응형 데몬 프로세스의 프로세스(600)의 하나의 실시예의 흐름도이다. 프로세스는, 하드웨어(예컨대, 회로, 전용 논리, 프로그래머블 논리, 마이크로코드 등), 소프트웨어(예컨대, 하드웨어 시뮬레이션을 수행하는 처리 디바이스에서 실행된 명령어들), 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 하나의 실시예에서, 프로세스(600)는, 도 3a 또는 3b의 데몬(306/316)과 같은, 적응형 데몬에 의해 수행된다. 도 6에서, 프로세스(600)는 메시지가 수신되는지를 결정하는 것으로 시작한다. 하나의 실시예에서, 프로세스(600)는 데몬 프로세스에 의해 사용되는 사용자 버퍼에 메시지가 존재하는지를 결정한다. 프로세스(600)에 대한 메시지가 없는 경우, 실행은 종료된다.
- [0061] 처리할 프로세스(600)에 대한 메시지가 있는 경우, 블록 604에서, 프로세스(600)는 메시지가 중요도에 대해 마크되어 있는지를 결정한다. 하나의 실시예에서, 프로세스(600)는, 전술한 바와 같은 중요도 비트와 같은, 중요도 플래그가 설정되어있는지를 체크한다. 메시지가 중요도에 대해 마크되어 있지 않은 경우, 실행 프로세스는 아래의 블록 608로 진행된다. 메시지가 중요도에 대해 마크되어 있는 경우, 프로세스(600)는 로컬 변수를 논제로 값으로 설정하고, 여기서, 로컬 변수는 블록 606에서 메시지가 중요한 것인지 아닌지 여부를 나타낸다.
- [0062] 프로세스(600)는 블록 608에서 메시지를 처리한다. 하나의 실시예에서, 프로세스(600)는 요청된 정보를 수집하는 액션을 수행한다. 예를 들어, 하나의 실시예에서, 프로세스(600)는 모바일 인터넷 공유 서비스의 상태를 결정하고, 전력 관리를 수행하고, 검색을 수행하고, 설치된 애플리케이션의 목록을 리턴하는 등을 할 수 있다. 프로세스(600)는, 블록 610에서, 요청된 정보와 함께 응답을 전송자에게 전송한다. 하나의 실시예에서, 프로세스(600)는, 도 3b에서 전술한 바와 같이, 사용자 애플리케이션 프로세스에 응답을 전송한다.
- [0063] 블록 612에서, 프로세스(600)는, 메시지가 중요도에 대해 마크되어 있는지를 결정한다. 하나의 실시예에서, 프로세스(600)는 중요도 플래그가 참(true)으로 설정되어 있는지를 체크한다. 메시지가 중요도에 대해 마크되어 있는 경우에, 블록 614에서, 프로세스(600)는 중요도 어썬션을 드롭하기 위해 커널 호출을 한다. 중요도에 대해 마크되어 있는 메시지가 없다는 중요도 플래그의 경우, 실행이 종료된다.
- [0064] 위의 도 6에서, 프로세스(600)는 중요도 어썬션을 드롭하기 위해 호출한다. 중요도 어썬션의 드롭은 데몬 프로세스의 중요도가 데몬 프로세스의 디폴트 중요도에 리턴되어야 하는지 여부를 결정하기 위해 커널에 의해 사용된다. 도 7은 중요도 어썬션을 드롭하는 커널의 프로세스(700)의 하나의 실시예의 흐름도이다. 프로세스는, 하드웨어(예컨대, 회로, 전용 논리, 프로그래머블 논리, 마이크로코드 등), 소프트웨어(예컨대, 하드웨어 시뮬레이션을 수행하는 처리 디바이스에서 실행된 명령어들) 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 하나의 실시예에서, 프로세스(700)는 도 3a 또는 3b의 커널(320)과 같은 커널에 의해 수행된다. 도 7에서, 프로세스(700)는, 블록 702에서, 현재 프로세스가 적응형 데몬인지를 결정하는 것으로 시작한다. 현재 프로세스가 적응형 데몬이 아닌 경우, 실행이 종료된다.
- [0065] 프로세스가 적응형 데몬인 경우, 프로세스(700)는, 블록 704에서, 현재 프로세스에 대한 "중요도 어썬션 카운트"를 감소시킨다. 이상에서 설명한 바와 같이, 중요도 어썬션 카운트는 프로세스에 대해 높아진 현재 어썬션의 수를 추적한다(keep track of). 프로세스(700)는 블록 706에서 중요도 어썬션 카운트가 제로인지를 결정한다. 제로인 중요도 어썬션 카운트는, 현재 프로세스에 대해 높아진 임의의 현재 어썬션이 존재하지 않음을 의미한다. 중요도 어썬션 카운트가 제로인 경우, 프로세스(700)는, 블록 708에서, 현재 프로세스를 디폴트 중요도로 디모트한다. 블록 706에서, 중요도 카운트가 제로가 아닌 경우, 실행이 종료된다.

- [0066] 도 8은 요구에 따라 서비스를 론치하는 프로세스(800)의 하나의 실시예의 흐름도이다. 프로세스는, 하드웨어(예컨대, 회로, 전용 논리, 프로그래머블 논리, 마이크로코드 등), 소프트웨어(예컨대, 하드웨어 시뮬레이션을 수행하는 처리 디바이스에서 실행된 명령어들) 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 하나의 실시예에서, 프로세스(800)는, 도 3a 또는 3b의 론치 서비스(326)와 같은, 론치 서비스에 의해 수행된다. 도 8에서, 프로세스(800)는, 블록 802에서, 프로세스(800)가 처리할 임의의 보류중인 이벤트가 있는지를 결정하여 시작한다. 보류중인 이벤트가 없는 경우, 프로세스(800)는 블록 802로 다시 진행되어 다른 보류중인 이벤트를 기다린다.
- [0067] 보류중인 이벤트가 있는 경우, 블록 804에서, 프로세스(800)는, 메시지 서비스가 생성되어야 한다는 것을 이벤트가 나타내는지를 결정한다. 하나의 실시예에서, 메시지 서비스는, 메시지 경로/접속의 구축, 메시지 교환, 및 그들 사이의 중요도 제공 속성을 관리하기 위해 데몬 및 사용자 애플리케이션 둘 다에 의해 사용되는 서비스들의 세트(예컨대, 라이브러리)이다. 그것은, 사용자 애플리케이션과 데몬 사이에서 정보를 교환하는 추상(abstract)을 제공하고, 추가로, 중요도 제공을 원활하게 처리한다. 예를 들어, 하나의 실시예에서, 메시지 서비스는 서비스 요청을 처리하고 실현하기 위해 메시지를 전송 및 수신할 수 있는, 검색 데몬과 같은, 데몬일 수 있다. 이벤트가, 블록 806에서, 메시지 서비스가 생성될 것임을 나타내는 경우, 프로세스(800)는 메시지 서비스에 대한 서비스 포트를 생성하고, 이 서비스 포트를 감시 목록(watch list)에 추가한다. 하나의 실시예에서, 서비스 포트를 생성하고, 이 포트를 감시 리스트에 추가함으로써, 프로세스(800)는 메시지가, 생성된 포트에 이 서비스를 위해 도착하는지를 결정할 수 있다. 프로세스(800)가 처리할 또 다른 이벤트를 기다림에 따라 실행 프로세스는 위의 블록 802로 진행된다.
- [0068] 블록 808에서, 이벤트가, 메시지 서비스가 생성될 것임을 나타내는 이벤트가 아닌 경우에, 프로세스(800)는 론치될 데몬을 위한 서비스 포트에 메시지가 있는지를 결정한다. 블록 810에서, 그러한 메시지가 있는 경우, 프로세스(800)는 데몬을 론치하고 이 데몬에 서비스 포트를 연관시키기 위해 커널을 호출한다. 하나의 실시예에서, 프로세스(800)는 서비스 포트에 대한 적절한 데몬을 결정하고 커널을 호출하여 해당 데몬을 론치한다. 또한, 프로세스(800)는 론치된 데몬에 메시지 서비스 포트를 연관시킨다. 본 실시예에서, 커널은 서비스 포트를 검사하고, 이 포트가 임의의 중요도 메시지를 갖는지 여부를 결정한다. 커널이 중요도 메시지가 있다고 결정한 경우, 커널은 더 높은 중요도로 데몬을 론치할 것이다. 서비스 포트에 중요도 메시지가 없는 경우, 커널은 그 데몬의 디폴트 중요도로 데몬을 론치한다. 게다가, 커널은 데몬/서비스 포트 연관성을 알기 때문에, 데몬이 정상/낮은 중요도로 론치되는 경우 및 나중에 중요도 메시지가 이 서비스 포트에서 큐되는 경우, 커널은 중요도 메시지를 큐잉하는 시간에 더 높은 중요도로 데몬을 프로모트할 수 있다. 프로세스(800)가 처리할 또 다른 이벤트를 기다림에 따라, 실행 프로세스는 위의 블록 802로 진행된다.
- [0069] 블록 808에서, 서비스 포트에 메시지가 없는 경우, 프로세스(800)는 이벤트가 블록 812에서 해당 데몬의 론치 이후에 적응형 데몬 체크인을 위한 것인지를 결정한다. 그렇다면, 블록 814에서, 프로세스(800)는 서비스 포트를 적응형 데몬으로 이전한다. 하나의 실시예에서, 이는 적응형 데몬이 해당 포트에서 메시지를 전송 및 수신하게 한다. 프로세스(800)가 처리할 또 다른 이벤트를 기다림에 따라, 실행은 블록 802로 진행된다. 블록 816에서, 이벤트가 적응형 데몬 체크인이 아닌 경우, 프로세스(800)는 다른 이벤트들을 처리한다.
- [0070] 전술한 바와 같이, 데몬 프로세스 중요도는, 데몬에 대해 인큐되는 중요도 제공자로부터의 메시지 또는 프로모트된 중요도로 론치되는 데몬을 트리거하는 중요도 제공자로부터의 메시지에 기초하여 실행중인 데몬에 대해 프로모트될 수 있다. 데몬 중요도가 프로모트될 때, 커널은 데몬에 대한 어썬션을 취한다. 이러한 데몬 프로세스는, 보류중인 어썬션이 제로로 드롭될 때 그의 중요도가 디모트될 수 있다. 어썬션은 중요도로 마크되는 메시지를 처리하는 데몬에 응답하여 데몬에 의해 드롭된다. 도 9는 적응형 데몬 프로세스 중요도를 프로모트 및 디모트하는 시스템(900)의 하나의 실시예의 블록도이다. 도 9에서, 시스템(900)은 사용자 애플리케이션 프로세스(902), 적응형 데몬 프로세스(904), 및 커널(906)을 포함한다. 하나의 실시예에서, 사용자 애플리케이션 프로세스(902)는 적응형 데몬 프로세스(904)와 메시지를 주고 받기 위해 메시지 서비스(940)를 사용한다. 본 실시예에서, 메시지 서비스(940)는, 사용자 애플리케이션 프로세스(902)와 적응형 데몬 프로세스(906) 사이에서 메시지 통신을 관리하기 위해, 사용자 애플리케이션 프로세스(902)와 적응형 데몬 프로세스(906)에 걸쳐 작동하는 서비스이다. 하나의 실시예에서, 메시지 서비스(940)는 동기 및 비동기 메시지 통신을 지원한다.
- [0071] 하나의 실시예에서, 사용자 애플리케이션 프로세스(904)는 적응형 데몬 프로세스(906)에 메시지를 전송하기 위해 메시지 서비스(940)를 사용한다. 예를 들어, 하나의 실시예에서, 사용자 애플리케이션 프로세스(902)는, 유틸리티 수면을 방지하고, 검색을 수행하는 등을 하기 위해 전력 어썬션을 보유하도록 개인용 핫스팟 서비스의 상태를 요청할 수 있다. 본 실시예에서, 사용자 애플리케이션 프로세스(902)는 메시지(920)를 생성하기 위해 사용

자 애플리케이션 접속(918)을 사용하여 메시지(922)를 전송한다. 이 메시지(920)는 커널(906)로 전송되는데, 여기서, 커널은 적응형 데몬 프로세스(904)에 대한 포트 큐(912)에 메시지(920)를 인큐하기 위해 커널 msg_send를 사용한다. 하나의 실시예에서, 포트 큐는 적응형 데몬 프로세스(904)와 통신된 메시지를 큐잉하기 위해 커널(906)에 의해 사용되는 큐의 세트이다. 하나의 실시예에서, 커널(906)의 커널 msg_send(914) 기능은 메시지에 대한 중요도(908)를 설정하고, 적응형 데몬 프로세스(904)에 대한 어썬션(910)을 취한다. 예를 들어, 하나의 실시예에서, 커널(906)은 위의 도 4에서 설명한 바와 같이 메시지 중요도 설정 및 어썬션 취득을 수행한다.

[0072] 포트 큐(912)에 인큐된 메시지로, 커널(906)은 포트 큐(912)로부터 메시지를 디큐하기 위해 기능 커널 msg_recv(916)을 수행하고, 그 메시지를 적응형 데몬 프로세스(904)를 위한 사용자 버퍼로 복사한다. 하나의 실시예에서, 커널(906)은 위의 도 5에서 설명된 바와 같이 메시지를 디큐하고 복사한다.

[0073] 적응형 데몬 프로세스(904)는 메시지 객체(938)로서 데몬 접속(932)을 통해 메시지(928)를 수신한다. 하나의 실시예에서, 적응형 데몬 프로세스(904)는 메시지(928)에 응답하지 않는다. 본 실시예에서, 메시지 서비스(940)는 메시지 객체(938)를 파괴한다. 메시지 객체 파괴(message object destruction; 934)에 응답하여, 메시지 서비스(940)는 커널로의 어썬션(926)을 드롭한다. 어썬션을 드롭함으로써, 커널(906)은 적응형 데몬 프로세스(904)가 적응형 데몬 프로세스(904)의 중요도를 디모트할지를 결정한다. 하나의 실시예에서, 커널(906)은 적응형 데몬 프로세스(904)가 위의 도 6에서 설명한 바와 같이 디모트될지를 결정한다.

[0074] 또 다른 실시예에서, 메시지 서비스는 수신된 메시지(928)에 대한 응답을 준비한다. 본 실시예에서, 메시지 서비스(940)는 메시지 객체(938)에 대한 응답 객체(936)를 생성한다. 응답 객체(936)를 생성함으로써, 메시지 객체(938)에 포함된 중요도는 응답 객체(936)로 이전된다. 메시지 서비스(940)는 데몬 접속(932)과 메시지(928)를 통해 그리고 커널(906)을 통해 사용자 애플리케이션 프로세스(904)에 응답을 전송한다. 또한, 응답을 전송함으로써, 메시지 서비스(940)는 또한 응답 객체(930)를 파괴한다. 응답 객체는 중요도 세트를 가지고 있기 때문에, 응답 객체의 파괴는 메시지 서비스(940)가 커널로의 어썬션(926)을 드롭하게 한다. 어썬션을 드롭함으로써, 커널(906)은 적응형 데몬 프로세스(904)가 적응형 데몬 프로세스(904)의 중요도를 디모트할지를 결정한다. 하나의 실시예에서, 커널(906)은 적응형 데몬 프로세스(904)가 위의 도 6에 설명된 바와 같이 디모트될지를 결정한다. 또한, 응답은 메시지 서비스(940)의 msg_recv 기능을 사용하는 사용자 애플리케이션 프로세스(902)에게 포트 큐(912)를 통해 메시지(828)를 사용하여 전송된다. 본 실시예에서, 메시지 서비스는 사용자 애플리케이션 프로세스(902)에 응답을 전달하기 위해 메시지(920) 및 사용자 애플리케이션 접속(918)을 사용한다.

[0075] 전술한 바와 같이, 사용자 애플리케이션 프로세스로부터의 메시지의 수신에 기초하여 데몬 프로세스의 중요도가 변경될 수 있다. 또 다른 실시예에서, 제2 데몬 프로세스는, 사용자 애플리케이션 프로세스로부터의 메시지를 전송하는 또 다른 데몬에 응답하여 중요도가 변경될 수 있다. 도 10은 제1 데몬에 의해 수신된 메시지에 기초하여 제2 데몬 중요도를 프로모트하는 프로세스(1000)의 하나의 실시예의 흐름도이다. 프로세스는 하드웨어(예컨대, 회로, 전용 논리, 프로그래머블 논리, 마이크로코드 등), 소프트웨어(예컨대, 하드웨어 시뮬레이션을 수행하는 처리 디바이스에서 실행된 명령어들) 또는 이들의 조합을 포함할 수 있는 처리 논리에 의해 수행될 수 있다. 하나의 실시예에서, 도 11의 커널(1132)과 같은 커널이 프로세스(1000)를 수행한다. 하나의 실시예에서, 사용자 애플리케이션 프로세스로부터 요청을 수신할 수 있고 또 다른 데몬으로부터 다른 정보를 요청할 수 있는 데몬이 프로세스(1000)를 수행할 것이다. 예를 들어, 하나의 실시예에서, 프로세스(1000)는 아래의 데몬(1106A/1116)의 메시지 처리 모듈(1128)에 의해 수행될 수 있다. 도 10에서, 프로세스(1000)는 블록 1002에서 제1 메시지를 수신하여 시작한다. 하나의 실시예에서, 이 메시지는 사용자 애플리케이션 프로세스로부터의 메시지와 같이 중요도를 운반하는 메시지이다. 또 다른 실시예에서, 메시지는 중요도 플래그 세트로 마크될 수 있다. 예를 들어, 하나의 실시예에서, 메시지는 도 9에서 전술한 바와 같이 중요도 비트 세트를 갖는다. 또 다른 실시예에서, 메시지는 또 다른 데몬에 의해 전송될 수 있고, 여기서, 커널은 중요도를 운반하는 것으로서 메시지를 마크한다. 본 실시예에서, 데몬으로부터의 것으로 중요도를 운반하는 것으로서 마크되는 메시지를 가짐으로써, 데몬 프로세스 우선 순위의 스트링은 프로모트되거나 디모트될 수 있다. 하나의 실시예에서, 아래의 도 10에서는, 데몬에 의한 이러한 메시지의 수신에 도 3a 및 4에서 전술된 바와 같이 프로모트된 데몬의 프로세스 중요도를 가질 것이다.

[0076] 블록 1004에서, 프로세스(1000)는 이 메시지에 대한 응답이 또 다른 데몬으로부터의 정보를 사용하는지를 결정한다. 예를 들어, 하나의 실시예에서, 검색 데몬은 검색 요청을 수신한다. 이 검색 데몬은, 검색 요청을 실현하기 위해, 모바일 디바이스에 설치된 애플리케이션의 목록을 결정할 필요가 있다. 검색 데몬은 이 목록을 얻기 위해 설치 데몬에 요청을 한다. 프로세스(1000)가 또 다른 데몬으로부터의 정보를 사용하지 않는 경우, 실

행은 아래의 블록 1010으로 진행된다.

- [0077] 프로세스(1000)가 또 다른 데몬으로부터의 정보를 사용하는 경우, 블록 1006에서, 프로세스(1000)는 이 다른 데몬에 제2 메시지를 전송하는데, 여기서, 커널은 이 메시지가 중요도를 운반하는 것으로서 마크한다. 또한, 커널은 다른 데몬의 중요도를 프로모트하고, 이 데몬에 대한 어썬션을 취하고, 이 다른 데몬에 대한 이 메시지를 큐잉한다. 블록 1008에서, 프로세스(1000)는 제2 메시지에 대한 응답을 수신한다. 블록 1010에서, 프로세스(1000)는 제1 메시지에 대한 응답을 준비하고 응답 메시지에 중요도를 마크한다. 또한, 프로세스(1000)는 블록 1002에서 제1 메시지를 수신한 데몬에 대한 어썬션을 드롭한다. 블록 1012에서, 프로세스(1000)는 제1 메시지를 전송한 프로세스에 응답을 다시 전송한다. 하나의 실시예에서, 이 메시지를 전송함으로써, 프로세스(1000)를 수행하는 데몬의 프로모트된 중요도가 디모트된다.
- [0078] 도 11은, 제2 데몬 프로세스(1106B)의 중요도가 프로모트된, 제2 데몬 프로세스(1106B)와 통신하는 제1 데몬 프로세스(1106A)를 갖는 모바일 디바이스(1100)의 하나의 실시예의 블록도이다. 도 11에서, 모바일 디바이스(1100)는 사용자 애플리케이션 프로세스(1104)와 데몬 프로세스(1106A-N)의 실행을 관리하는 운영 체제(1102)를 포함한다. 또한, 사용자 애플리케이션 프로세스(1104)는 높은 중요도로 실행하고, 데몬 프로세스(1106A-N)는 처음에는 낮은 중요도로 실행한다. 사용자 애플리케이션 프로세스(1104)가 데몬 프로세스(예컨대, 데몬 프로세스(1106A))로부터 서비스를 요청할 때, 이 데몬 프로세스는 이 서비스 요청을 처리하기 위해 더 높은 중요도로 프로모트된다. 하나의 실시예에서, 이 데몬(1106A)은 적응형 데몬이라고 불린다. 예를 들어, 하나의 실시예에서, 사용자 애플리케이션 프로세스(1104)는 메시지(1112)를 통해 데몬(1106A)에 서비스를 요청한다. 메시지를 수신하는 데몬(1106A)에 응답하여, 데몬의 중요도는 더 높은 중요도의 데몬(1116)으로 프로모트된다. 하나의 실시예에서, 프로모트된 데몬(1116)의 중요도는 요청하는 사용자 애플리케이션 프로세스(1104)와 동일한 한편, 대안적인 실시예에서, 데몬(1116)의 프로모트된 프로세스는 요청하는 사용자 애플리케이션 프로세스(1104)의 중요도보다 더 높거나 더 낮을 수 있다.
- [0079] 데몬 A 프로세스(1116)가 메시지를 수신하고 중요도가 프로모트되는 것에 응답하여, 데몬 A 프로세스는 데몬 B 프로세스(1106B)에게 또 다른 메시지(1136)를 전송한다. 이 메시지(1136)는 데몬 A에 전송된 것과 동일한 메시지일 수 있거나 다른 메시지일 수 있다. 메시지(1136)를 수신하는 데몬(1106B)에 응답하여, 데몬 B의 중요도가 더 높은 중요도의 데몬 B(1120)로 프로모트된다. 하나의 실시예에서, 프로모트된 데몬 B(1120)의 중요도는 초기 요청하는 사용자 애플리케이션 프로세스(1104)와 동일한 한편, 대안적인 실시예에서, 프로모트된 데몬(1120)의 프로세스는 요청하는 사용자 애플리케이션 프로세스(1104)의 중요도보다 더 높거나 더 낮을 수 있다.
- [0080] 본 실시예에서, 프로모트된 데몬 B(1120)는 제2 메시지를 처리하고 데몬 A(1116)에 응답(1122)을 전송한다. 또한, 데몬 B 프로세스의 중요도는 데몬 B(1106B)에 대한 디폴트 중요도로 디모트된다. 하나의 실시예에서, 데몬 프로세스 B(1120)는 어썬션을 취하고, 커널은 데몬 B 프로세스(1120)의 중요도를 디모트할 수 있다. 응답(1122)을 수신하는 것에 응답하여, 데몬 A는 초기 메시지(1112)에 대한 응답을 형성하고, 사용자 애플리케이션 프로세스(1104)로 그 메시지를 전송한다. 또한, 데몬 A 프로세스(1116)의 중요도는 이 프로세스에 대한 디폴트 중요도로 디모트된다. 하나의 실시예에서, 운영 체제 커널(1132)은 데몬 프로세스 프로모션 및 디모션을 처리한다. 또 다른 실시예에서, 운영 체제 커널(1132)은 메시지 서비스(1134)를 사용하여 데몬 프로세스 프로모션과 디모션을 처리한다. 본 실시예에서, 프로세스 관리(1134)는 사용자 애플리케이션 프로세스(1104), 데몬 프로세스(1106A/1116), 및 데몬 프로세스(1106B/1120) 사이의 메시지의 전송 및 수신을 처리한다. 이러한 프로세스들 사이에서 메시지 전달을 처리함으로써, 프로세스 관리(1132)는 데몬 프로세스 프로모션과 디모션을 관리할 수 있다.
- [0081] 예를 들어, 하나의 실시예에서, 검색 데몬을 사용하여 검색이 실시되는 예를 고려한다. 이러한 예에서, 검색 데몬은 사용자 애플리케이션 프로세스로부터의 검색 요청의 수신 결과로서 프로모트된 검색 데몬 프로세스 중요도를 가질 수 있는 적응형 데몬이다. 검색 요청 메시지의 수신에 응답하여, 검색 데몬은 모바일 디바이스에 설치된 애플리케이션의 목록을 얻기 위해 설치 데몬에 제2 메시지를 전송한다. 중요도 프로모션 자체를 취득한 검색 데몬은, 설치된 애플리케이션의 목록에 대한 제2 메시지를 설치 데몬으로 전송함으로써, 이 중요도를 제공한다. 이 예에서, 설치 데몬은 검색 데몬으로부터 제2 메시지의 수신 시에 프로모트된 중요도를 가질 것이다. 또한, 설치 데몬의 중요도는, 설치 데몬이 검색 데몬에 응답을 전송할 때, 설치 데몬의 디폴트 중요도 디모트된다. 하나의 실시예에서, 메시지 처리 및 데몬 프로세스 프로모션/디모션은, 전송한 메시지 서비스(1034)와 같은, 메시지 서비스에 의해 처리된다.
- [0082] 도 12는 프로세스들 사이에서 메시지의 전송 및 수신을 관리하는 프로세스 관리 모듈(322)이다. 하나의 실시예

에서, 프로세스 관리 모듈(322)은, 도 3a-b의 커널(320) 및/또는 도 10의 커널(1032)과 같이, 메시지의 전송 및 수신을 용이하게 하기 위해 커널에 의해 사용된다. 하나의 실시예에서, 프로세스 관리 모듈(322)은 메시지 전송 모듈(1202), 메시지 수신 모듈(1204), 및 드롭 어썬션 중요도 모듈(1206)을 포함한다. 본 실시예에서, 메시지 전송 모듈(1202)은 위의 도 4에서 설명한 바와 같이 메시지를 전송한다. 메시지 수신 모듈(1204)은 위의 도 5에 설명된 바와 같이 수신한다. 드롭 어썬션 중요도 모듈(1206)은 위의 도 6에 설명된 바와 같이 어썬션을 드롭한다.

[0083] 도 13은 데몬으로 향하는 메시지에 기초하여 그 데몬 중요도를 프로모트하는데 사용되는 메시지를 전송하는 메시지 전송 모듈(1202)이다. 하나의 실시예에서, 메시지 전송 모듈(1202)은 프로세스 중요도 제공자 모듈(1302), 프로세스 중요도 수신자 모듈(1304), 메시지 마크 모듈(1306), 중요도가 낮은 프로세스 모듈(1308), 프로세스 프로모트 모듈(1310), 및 메시지 큐 모듈(1312)을 포함한다. 하나의 실시예에서, 프로세스 중요도 제공자 모듈(1302)은 현재 프로세스가 도 4의 블록 402에서 전송한 바와 같은 중요도 제공자인지를 결정한다. 프로세스 중요도 수신자 모듈(1304)은 대상 프로세스가 도 4의 블록 404에서 전송된 바와 같은 중요도 수신자인지를 결정한다. 메시지 마크 모듈(1306)은 메시지에 마크하고 도 4의 블록 406에 전송된 바와 같이 중요도 어썬션 카운트를 증가시킨다. 중요도가 낮은 프로세스 모듈(1308)은 대상 프로세스가 도 4의 블록 408에 전송된 바와 같이 더 낮은 중요도를 갖는지를 결정한다. 프로세스 프로모트 모듈(1310)은 도 4의 블록 410에서 전송된 바와 같이 대상 프로세스 중요도를 변경한다. 메시지 큐 모듈(1312)은 도 4의 블록 413에 전송된 바와 같이 포트에 메시지를 큐잉한다.

[0084] 도 14는 데몬으로 향하는 메시지에 기초하여 그 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 메시지 수신 모듈(1204)이다. 하나의 실시예에서, 메시지 수신 모듈(1204)은, 큐잉된 메시지 모듈(1402), 디큐 메시지 모듈(1404), 메시지 마크 모듈(1406), 프로세스 중요도 수신자 모듈(1408), 플래그 제거 모듈(1410), 및 메시지 복사 모듈(1412)을 포함한다. 하나의 실시예에서, 큐잉된 메시지 모듈(1402)은 도 5의 블록 502에 전송된 바와 같이 큐잉된 메시지가 존재하는지를 결정한다. 디큐 메시지 모듈(1404)은 도 5의 블록 504에서 전송된 바와 같이 포트로부터 메시지를 디큐한다. 메시지 마크 모듈(1406)은 도 5의 블록 506에 전송된 바와 같이 메시지에 중요도가 마크되어 있는지를 결정한다. 프로세스 중요도 수신자 모듈(1408)은 도 5의 블록 508에 전송된 바와 같이 현재 프로세스가 적응형 데몬 프로세스인지를 결정한다. 플래그 제거 모듈(1410)은 도 5의 블록 510에 전송된 바와 같이 플래그를 제거하고 중요도 카운트를 감소시킨다. 메시지 복사 모듈(1412)은 도 5의 블록 512에서 전송한 바와 같이 사용자 버퍼에 메시지를 복사한다.

[0085] 도 15는 데몬 중요도를 프로모트하는데 사용되는 메시지를 수신하는 데몬 msg_recv 모듈(324)이다. 하나의 실시예에서, 데몬 msg_recv 모듈(324)은 수신된 메시지 모듈(1502), 마크된 메시지 모듈(1504), 메시지 중요도 설정 모듈(1506), 메시지 프로세스 모듈(1508), 메시지 응답 모듈, 중요도 메시지 모듈, 및 드롭 중요도 모듈(1514)을 포함한다. 본 실시예에서, 수신된 메시지 모듈(1502)은 도 6의 블록 602에 설명된 바와 같이 메시지가 수신되는지를 결정한다. 마크된 메시지 모듈(1504)은 도 6의 블록 604에 설명된 바와 같이 메시지에 중요도에 대해 마크한다. 메시지 중요도 설정 모듈(1506)은 도 6의 블록 606에 설명된 바와 같이 중요도를 참으로 설정한다. 메시지 프로세스 모듈(1508)은 도 6의 블록 608에서 설명된 바와 같이 메시지를 처리하고 필요한 액션을 수행한다. 메시지 응답 모듈(1510)은 도 6의 블록 610에 설명된 바와 같이 응답을 전송한다. 중요도 메시지 모듈(1512)은 도 6의 블록 612에 설명된 바와 같이 메시지가 중요한 것으로 마크되는지를 결정한다. 중요도 드롭 모듈(1514)은 도 6의 블록 614에 설명된 바와 같이 중요도 어썬션을 드롭한다.

[0086] 도 16은 중요도 어썬션을 드롭하는 중요도 어썬션 드롭 모듈(1206)이다. 하나의 실시예에서, 중요도 어썬션 드롭 모듈(1206)은 현재 프로세스 모듈(1602), 어썬션 카운트 감소 모듈(1604), 어썬션 카운트 제로 모듈(1606), 프로세스 디모션 모듈(1608)을 포함한다. 본 실시예에서, 현재 프로세스 모듈(1602)은 현재 프로세스가 도 7의 블록 702에 전송된 바와 같이 중요도 프로세스인지를 결정한다. 어썬션 카운트 감소 모듈(1604)은 도 7의 블록 704에서 전송한 바와 같이 중요도 어썬션 카운트를 감소시킨다. 어썬션 카운트 제로 모듈(1606)은 도 7의 블록 706에서 전송한 바와 같이 중요도 어썬션 카운트를 결정한다. 프로세스 디모션 모듈(1608)은 도 7의 블록 708에 전송된 바와 같이 현재 프로세스를 디모트한다.

[0087] 도 17은 요구에 따라 서비스를 론치하는 론치 서비스(326)이다. 하나의 실시예에서, 론치 서비스는 보류중인 이벤트 모듈(1702), 메시지 서비스 모듈(1704), 서비스 포트 생성 모듈(1706), 서비스 포트 메시지 모듈(1708), 적응형 데몬 모듈(1710), 론치 적응형 데몬(1712), 론치 데몬 모듈(1714), 적응형 데몬 체크인 모듈(1716), 서비스 포트 이전 모듈(1718) 및 다른 이벤트 프로세스 모듈(1720)을 포함한다. 본 실시예에서, 보류중인 이벤트 모듈(1702)은 위의 도 8의 블록 802에 설명된 바와 같이 임의의 보류중인 이벤트가 존재하는지를 결정한다. 메

시지 서비스 모듈(1704)은, 도 8의 블록 804에 전술된 바와 같이, 생성될 메시지 서비스가 존재하는지를 결정한다. 서비스 포트 생성 모듈(1706)은, 도 8의 블록 806에 전술된 바와 같이, 서비스 포트를 생성하고 이 서비스 포트를 감시 목록에 추가한다. 서비스 포트 메시지 모듈(1708)은, 도 8의 블록 808에 전술된 바와 같이, 서비스 포트에 메시지가 존재하는지를 결정한다. 적응형 데몬 모듈(1710)은, 도 8의 블록 810에 전술된 바와 같이, 메시지가 적응형 데몬을 위한 것인지 여부를 결정한다. 론치 적응형 데몬(1712)은, 도 8의 블록 812에 전술된 바와 같이, 적응형 데몬을 론치한다. 론치 데몬 모듈(1714)은, 도 8의 블록 814에 전술된 바와 같이, 데몬을 론치한다. 적응형 데몬 체크인 모듈(1716)은, 도 8의 블록 816에 전술된 바와 같이, 적응형 데몬이 론치 이후에 체크인되었는지를 결정한다. 서비스 포트 이전 모듈(1718)은, 도 8의 블록 818에 전술된 바와 같이, 적응형 데몬으로 서비스 포트를 이전한다. 다른 이벤트 프로세스 모듈(1720)은, 도 8의 블록 820에 전술된 바와 같이, 다른 이벤트를 처리한다.

[0088] 도 18은 제1 데몬에 의해 수신된 메시지에 기초하여 제2 데몬 중요도를 프로모트하는 메시지 처리 모듈(1508)이다. 하나의 실시예에서, 메시지 처리 모듈(1508)은 메시지 수신 모듈(1802), 데몬 메시지 모듈(1804), 데몬 메시지 전송 모듈(1806), 데몬 응답 수신 모듈(1808), 응답 준비 모듈(1810), 및 응답 전송 모듈(1812)을 포함한다. 본 실시예에서, 메시지 수신 모듈(1802)은, 도 9의 블록 902에 전술된 바와 같이, 제1 메시지를 수신한다. 데몬 메시지 모듈(1804)은, 도 9의 블록 904에 전술된 바와 같이, 제1 메시지 응답이 또 다른 데몬으로부터 정보를 필요로 하는지를 결정한다. 데몬 메시지 전송 모듈(1806)은, 도 9의 블록 906에 전술된 바와 같이, 제2 메시지를 다른 데몬으로 전송한다. 데몬 응답 수신 모듈(1808)은, 도 9의 블록 908에 전술된 바와 같이, 응답을 수신한다. 응답 준비 모듈(1810)은, 도 9의 블록 910에 전술된 바와 같이 제1 메시지로의 응답 메시지를 준비한다. 응답 전송 모듈(1812)은, 도 9의 블록 912에 전술된 바와 같이, 응답을 전송한다.

[0089] 도 19는 본 발명의 하나의 실시예에 사용할 수 있는 데이터 처리 시스템(1900)의 하나의 예를 보여준다. 예를 들어, 시스템(1900)은 도 3a-b에 도시된 바와 같이 디바이스(300)를 포함하여 구현될 수 있다. 도 19는 컴퓨터 시스템의 다양한 컴포넌트들을 도시하지만, 그러한 상세가 본 발명에 밀접한 관계가 없는 경우 컴포넌트들을 상호접속하는 임의의 특정한 아키텍처 또는 방식을 표현하기 위한 것이 아님을 유의한다. 또한, 더 적은 컴포넌트 또는 아마도 더 많은 컴포넌트를 갖는, 네트워크 컴퓨터 및 다른 데이터 처리 시스템 또는 다른 소비자 전자 디바이스가 본 발명에 사용될 수 있다는 것이 이해될 것이다.

[0090] 도 19에 도시된 바와 같이, 데이터 처리 시스템의 형태인 컴퓨터 시스템(1900)은 마이크로 프로세서(들)(1905)와 ROM(판독 전용 메모리)(1907) 및 휘발성 RAM(1909) 및 비-휘발성 메모리(1911)에 결합되는 버스(1903)를 포함한다. 마이크로 프로세서(1905)는 메모리(1907, 1909, 1911)로부터 명령어를 검색하고, 위에서 설명한 동작들을 수행하기 위해 명령어들을 실행할 수 있다. 버스(1903)는 이러한 다양한 컴포넌트들을 함께 상호접속하고, 또한 이러한 컴포넌트들(1905, 1907, 1909 및 1911)을 표시 제어기와 표시 디바이스(1913)에, 그리고 이를테면 마우스, 키보드, 모뎀, 네트워크 인터페이스, 프린터 및 본 기술 분야에서 잘 알려진 다른 디바이스들일 수 있는 입/출력(I/O) 디바이스와 같은 주변 디바이스에 상호접속한다. 일반적으로, 입/출력 디바이스(1915)는 입/출력 제어기(1913)를 통해 시스템에 결합된다. 휘발성 RAM(랜덤 액세스 메모리)(1909)은 일반적으로 메모리에서 데이터를 새로 고치거나 유지하기 위해 지속적으로 전력을 필요로 하는 동적 RAM(DRAM)으로서 구현된다.

[0091] 대용량 스토리지(1911)는 일반적으로 마그네틱 하드 드라이브 또는 광 자기 드라이브 또는 광 드라이브 또는 DVD RAM 또는 플래시 메모리 또는 다른 유형의 메모리 시스템으로서, 시스템으로부터 전력이 제거된 후에도 데이터(예컨대, 대량의 데이터)를 유지한다. 일반적으로, 대용량 스토리지(1911)는 또한 필수 사항은 아니지만 랜덤 액세스 메모리일 것이다. 도 19는, 대용량 스토리지(1911)가 데이터 처리 시스템의 컴포넌트들의 나머지에 직접 결합된 로컬 디바이스를 도시하지만, 본 발명은, 이를테면, 모뎀, 이더넷 인터페이스 또는 무선 네트워크와 같은 네트워크 인터페이스를 통해 데이터 처리 시스템에 결합되어 있는 네트워크 스토리지 디바이스와 같은 시스템으로부터 원격인 비-휘발성 메모리를 이용할 수 있다는 것이 이해될 것이다. 버스(1903)는 본 기술 분야에서 잘 알려진 다양한 브리지, 제어기 및/또는 어댑터를 통해 서로 접속된 하나 이상의 버스를 포함할 수 있다.

[0092] 도 20은 본 발명의 하나의 실시예에 사용할 수 있는 또 다른 데이터 처리 시스템(2000)의 예를 도시한다. 예를 들어, 시스템(2000)은, 도 3a-b에 도시된 바와 같이, 디바이스(300)로서 구현될 수 있다. 도 20에 도시된 데이터 처리 시스템(2000)은 처리 시스템(2011)을 포함하는데, 하나 이상의 마이크로프로세서일 수도 있고, 또는 시스템 온 칩 집적 회로일 수도 있으며, 시스템은 또한 처리 시스템에 의한 실행을 위해 데이터 및 프로그램을 저장하기 위한 메모리(2001)를 포함한다. 시스템(2000)은 또한 오디오 입/출력 서브시스템(2005)을 포함하는데,

예를 들어, 음악을 재생하거나 스피커와 마이크를 통해 전화 기능을 제공하기 위해 마이크 및 스피커를 포함할 수 있다.

[0093] 표시 제어기 및 표시 디바이스(2009)는 사용자를 위한 시각적 사용자 인터페이스를 제공하고; 이 디지털 인터페이스는 OS X 운영 체제 소프트웨어를 실행할 때 매킨토시 컴퓨터에 표시되거나 iOS 운영 체제를 실행할 때 애플 아이폰에 표시되는 것 등과 유사한 그래픽 사용자 인터페이스를 포함할 수 있다. 시스템(2000)은 또한 도 20의 시스템(2000)과 같은 또 다른 데이터 처리 시스템과 통신하기 위해 하나 이상의 무선 송수신기(2003)를 포함한다. 무선 송수신기는 WLAN 송수신기, 적외선 송수신기, 블루투스 송수신기 및/또는 무선 휴대 전화 송수신기일 수 있다. 도시되지는 않지만, 특정 실시예에서, 추가의 컴포넌트 또한 시스템(2000)의 일부일 수 있고, 특정 실시예에서는, 도 20에 도시된 것보다 더 적은 컴포넌트가 또한 데이터 처리 시스템에서 사용될 수 있다는 것이 이해될 것이다. 시스템(2000)은 또한 도 15의 시스템(1500)과 같은 또 다른 데이터 처리 시스템과 통신하는 하나 이상의 통신 포트(2017)를 포함한다. 통신 포트는 USB 포트, FireWire 포트, 블루투스 인터페이스 등일 수 있다.

[0094] 데이터 처리 시스템(2000)은 또한 사용자가 시스템에 입력을 제공할 수 있도록 제공되는 하나 이상의 입력 디바이스(2013)를 포함한다. 이러한 입력 디바이스는 키패드 또는 키보드 또는 터치 패널 또는 멀티 터치 패널일 수 있다. 데이터 처리 시스템(2000)은 또한 독(dock)을 위한 커넥터일 수 있는 선택적인 입/출력 디바이스(2015)를 포함한다. 도시되지는 않지만, 하나 이상의 버스가 이 분야에서 잘 알려진 바와 같은 다양한 컴포넌트들을 상호접속하는데 사용될 수 있다는 것이 이해될 것이다. 도 20에 도시된 데이터 처리 시스템은 핸드헬드 컴퓨터 또는 PDA(personal digital assistant), 또는 PDA 같은 기능성을 갖는 셀룰러 전화, 또는 아이팟과 같은 미디어 플레이어 또는 셀룰러 전화를 포함하는 핸드헬드 컴퓨터, 또는 이러한 디바이스들의 측면들이나 기능들을 조합하는 디바이스들, 이를테면, 하나의 디바이스 혹은 내장된 디바이스 혹은 다른 소비자 전자 디바이스에 PDA와 셀룰러 전화가 조합된 미디어 플레이어일 수 있다. 다른 실시예에서, 데이터 처리 시스템(2000)은 네트워크 컴퓨터 또는 또 다른 디바이스 내에 내장된 처리 디바이스, 또는 다른 유형의 데이터 처리 디바이스일 수 있는데, 도 20에 도시된 것보다 더 적은 컴포넌트들 또는 아마도 더 많은 컴포넌트들을 갖는다.

[0095] 본 발명의 적어도 특정 실시예는 휴대용 음악 및/또는 비디오 미디어 플레이어와 같은 디지털 미디어 플레이어의 일부일 수 있는데, 미디어를 제시하는 미디어 처리 시스템, 미디어를 저장하는 저장 디바이스를 포함할 수 있고, 안테나 시스템 및 미디어 처리 시스템과 결합된 무선 주파수(RF) 송수신기(예컨대, 셀룰러 전화용 RF 송수신기)를 더 포함할 수 있다. 특정 실시예에서, 원격 저장 디바이스에 저장된 미디어는 RF 송수신기를 통해 미디어 플레이어에 전송될 수 있다. 미디어는, 예를 들어, 음악이나 다른 오디오, 스틸 픽처, 또는 모션 픽처 중 하나 이상일 수 있다.

[0096] 휴대용 미디어 플레이어는, 캘리포니아주, 쿠파티노, 애플사의 아이팟® 또는 아이팟 나노® 미디어 플레이어의 클릭 휠 입력 디바이스와 같은 미디어 선택 디바이스, 터치 스크린 입력 디바이스, 푸시버튼 디바이스, 이동가능 포인팅 입력 디바이스 또는 다른 입력 디바이스를 포함할 수 있다. 미디어 선택 디바이스는 저장 디바이스 및/또는 원격 저장 디바이스에 저장된 미디어를 선택하는데 사용될 수 있다. 휴대용 미디어 플레이어는, 적어도 특정 실시예에서, 입력 디바이스를 통해 선택되고, 스피커나 이어폰(들)을 통해, 또는 표시 디바이스 상에, 또는 표시 디바이스와 스피커나 이어폰(들) 둘 다에 제시되는 미디어의 제목 또는 다른 표시자를 표시하는 미디어 처리 시스템에 결합되는 표시 디바이스를 포함할 수 있다. 휴대용 미디어 플레이어의 예는, 이들 둘 다 참조에 의해 여기에 통합된, 미국 특허 제 7,345,671호 및 미국 공개 특허 제2004/0224638호에 설명되어 있다.

[0097] 전술한 것의 일부는 전용 논리 회로와 같은 논리 회로 또는 마이크로컨트롤러 또는 프로그램 코드 명령어들을 실행하는 다른 형태의 처리 코어로 구현될 수 있다. 따라서, 전술한 프로세스들은 특정 기능을 수행하기 위해 기계가 이러한 명령어들을 실행하게 하는 기계-실행가능한 명령어들과 같은 프로그램 코드로 수행될 수 있다. 이러한 맥락에서, "기계"는, 중간형(또는 "추상") 명령어들을 프로세서 고유의 명령어(예컨대, "가상 머신"(예컨대, 자바 가상 머신), 인터프리터, 공통 언어 런타임, 하이-레벨 언어 가상 기계 등과 같은 추상적인 실행 환경)로 변환하는 기계, 및/또는 범용 프로세서 및/또는 특수 목적 프로세서와 같은 명령어들을 실행하도록 설계된 반도체 칩에 배치된 전자 회로(예컨대, 트랜지스터로 구현된 "논리 회로")일 수 있다. 전술한 프로세스들은 또한 프로그램 코드의 실행 없이 프로세스들(또는 그들의 일부)을 (기계의 대안으로 또는 기계와 조합하여) 수행하도록 설계된 전자 회로에 의해 수행될 수 있다.

[0098] 본 발명은 또한 여기에 설명된 동작들을 수행하기 위한 장치에 관한 것이다. 이 장치는 특별히 필요한 목적을 위해 구성될 수도 있고, 또는 컴퓨터에 저장된 컴퓨터 프로그램에 의해 선택적으로 활성화되거나 재구성된 범용

컴퓨터를 포함할 수 있다. 이러한 컴퓨터 프로그램은 컴퓨터 판독가능한 저장 매체, 이를테면, 이로 제한되는 것은 아니지만, 플로피 디스크, 광 디스크, CD-ROM 및 광 자기 디스크를 포함하는 임의의 유형의 디스크, 판독 전용 메모리(ROM), RAM, EPROM, EEPROM, 자기 또는 광 카드, 또는 전자적 명령어들을 저장하기에 적합하고, 각각 컴퓨터 시스템 버스에 결합된 임의의 유형의 매체에 저장될 수 있다.

[0099] 기계 판독가능 매체는 기계(예컨대, 컴퓨터)로 판독가능한 형태로 정보를 저장하거나 전송하는 임의의 메커니즘을 포함한다. 예를 들어, 기계 판독가능 매체는 판독 전용 메모리("ROM"); 랜덤 액세스 메모리("RAM"); 자기 디스크 저장 매체; 광 저장 매체; 플래시 메모리 디바이스 등을 포함한다.

[0100] 제조자의 아티클은 프로그램 코드를 저장하는데 사용될 수 있다. 프로그램 코드를 저장하는 제조자의 아티클은, 이로 제한되는 것은 아니지만, 하나 이상의 메모리(예컨대, 하나 이상의 플래시 메모리, (정적, 동적 또는 다른) 랜덤 액세스 메모리), 광 디스크, CD-ROM, DVD ROM, EPROM, EEPROM, 자기 또는 광 카드 또는 전자적 명령어들을 저장하기에 적절한 다른 유형의 기계-판독가능한 매체로서 구현될 수 있다. 프로그램 코드는 (예컨대, 통신 링크(예컨대, 네트워크 접속)를 통해) 전과 매체에 내장된 데이터 신호의 방식에 의해 원격 컴퓨터(예컨대, 서버)로부터 요청하는 컴퓨터(예컨대, 클라이언트)로 다운로드될 수 있다.

[0101] 전술한 상세한 설명은 컴퓨터 메모리 내의 데이터 비트에 대한 동작들의 알고리즘들 및 상징적 표현들의 측면에서 제시된다. 이러한 알고리즘적 설명들 및 표현들은 다른 당업자들에게 그들의 작업의 실체를 가장 효과적으로 전달하기 위해 데이터 처리 기술의 당업자들에 의해 사용되는 도구이다. 알고리즘은 여기에서는 일반적으로 원하는 결과로 이어지는 일관성 있는 시퀀스의 동작들로 간주된다. 동작들은 물리적 양의 물리적 조작을 필요로 하는 것이다. 보통, 필수적인 것은 아니지만, 이러한 양은 저장, 전송, 조합, 비교, 및 그렇지 않으면 조작될 수 있는 전기 또는 자기 신호의 형태를 취한다. 일반적인 사용의 이유로, 때로는, 이러한 신호들을 비트(bits), 값(values), 요소(elements), 기호(symbols), 문자(characters), 텀(terms), 숫자(numbers) 등으로 지칭하는 것이 편리하다는 것이 입증되었다.

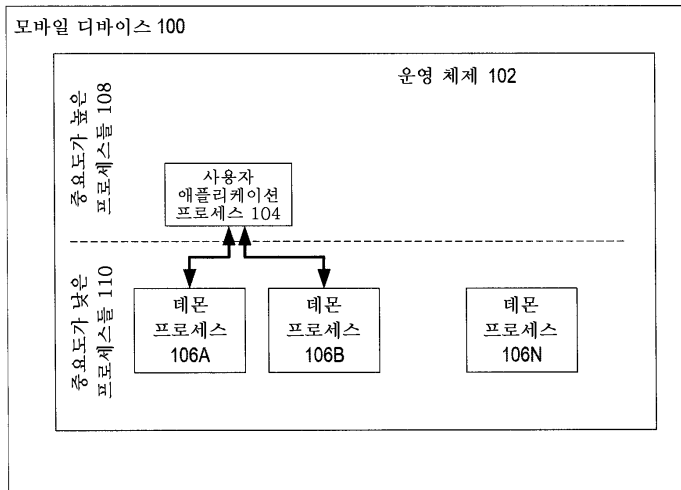
[0102] 그러나, 이들 및 유사한 용어 모두 적절한 물리량과 연관될 것이고, 단지 이러한 양에 적용되는 편리한 라벨이라는 것을 명심해야 한다. 달리 구체적으로 명시되지 않는 한, 전술로부터 분명한 바와 같이, 설명 전체에 걸쳐, "수신하는(receiving)", "마크하는(making)", "처리하는(processing)", "전송하는(sending)", "프로모트하는(promoting)", "디모트하는(demoting)", "리턴하는(returning)", "컴퓨팅하는(computing)", "인큐하는(enqueuing)", "이전하는(transferring)", "론치하는(launching)", "포워드하는(forwarding)", 감소시키는(decrementing)", "유지하는(maintaining)" 등과 같은 용어를 사용하는 논의는, 컴퓨터 시스템의 레지스터 및 메모리 내에 물리(전자)량으로 표현된 데이터를 조작하여 컴퓨터 시스템 메모리 또는 레지스터 또는 다른 그러한 정보 저장, 전송 또는 표시 디바이스 내에 물리량으로 유사하게 표현된 다른 데이터로 변형하는 컴퓨터 시스템 또는 유사한 전자적 컴퓨팅 디바이스의 액션 및 프로세스를 지칭한다.

[0103] 여기에 제시된 프로세스 및 디스플레이는 본질적으로 임의의 특정한 컴퓨터 나 다른 장치에 관련된 것이 아니다. 다양한 범용 시스템은 여기에서의 가르침에 따라 프로그램과 사용될 수 있고, 또는 설명된 동작들을 수행하기 위해 더 전문적인 장치를 구성하는 것이 편리하다는 것이 입증될 수 있다. 다양한 이러한 시스템에 대해 요구된 구조는 아래 설명으로부터 명백해질 것이다. 또한, 본 발명은 임의의 특정한 프로그래밍 언어를 기준으로 설명되지 않는다. 여기에 설명된 바와 같이 다양한 프로그래밍 언어가 본 발명의 가르침을 구현하는데 사용될 수 있다는 것이 이해될 것이다.

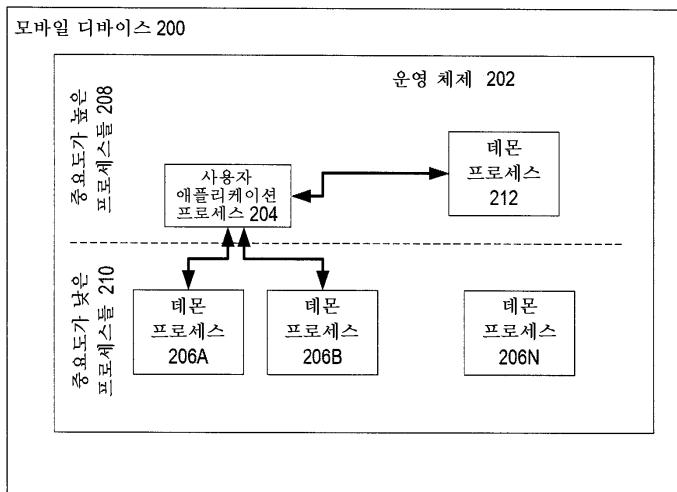
[0104] 전술한 논의는 단지 본 발명의 일부 예시적인 실시예를 설명한다. 당업자는 이러한 논의, 첨부 도면 및 청구항들로부터 본 발명의 사상 및 범위를 벗어나지 않고 다양한 변경이 실시될 수 있다는 것을 쉽게 인식할 것이다.

도면

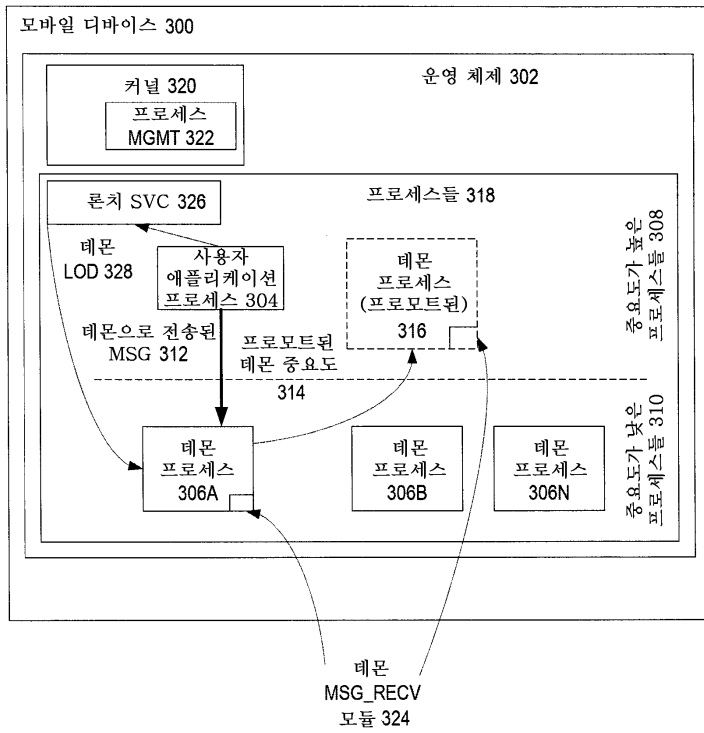
도면1



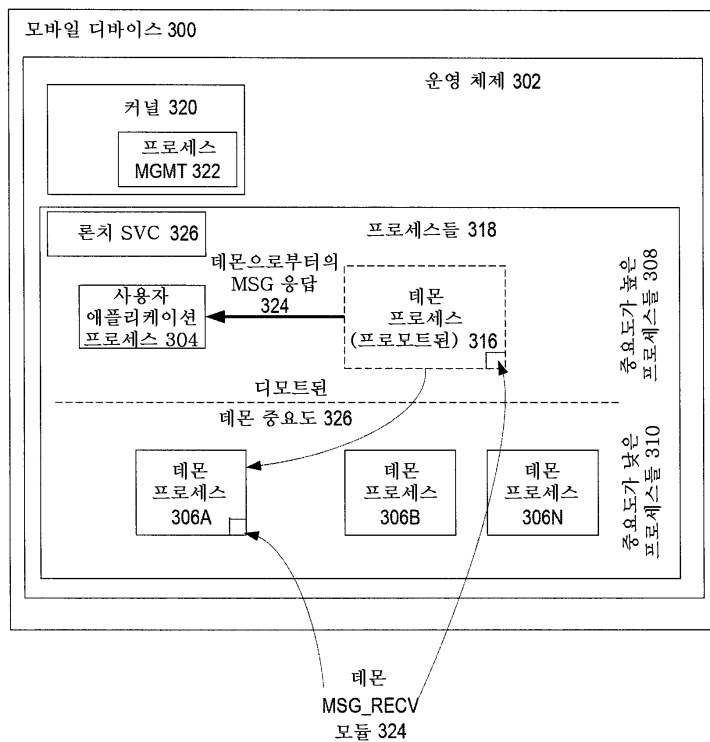
도면2



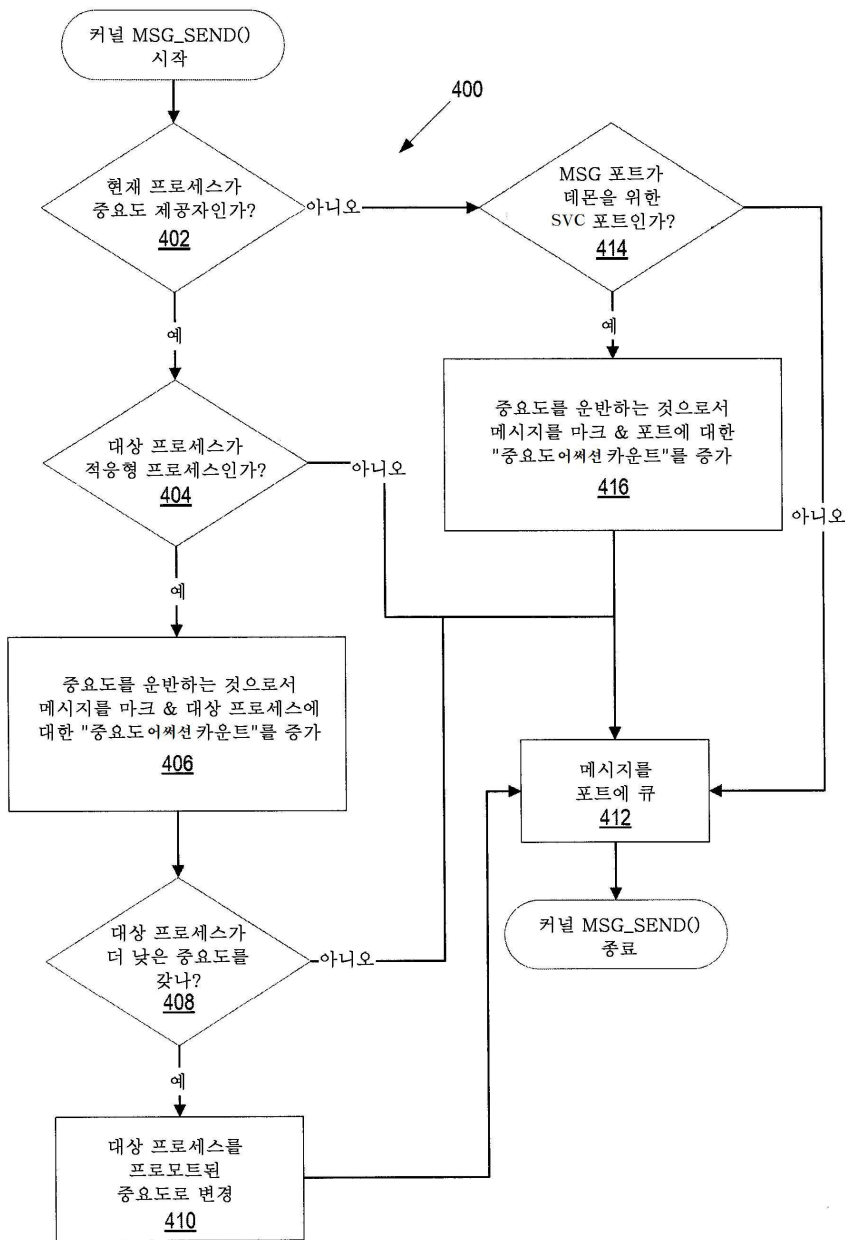
도면3a



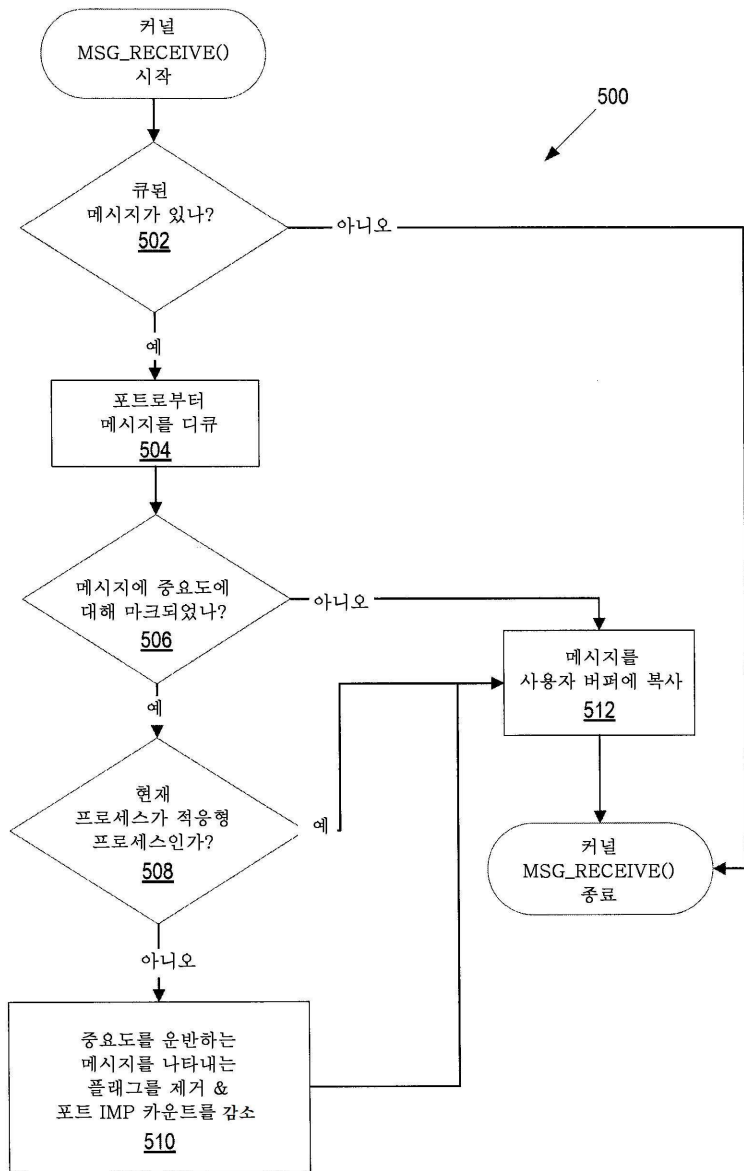
도면3b



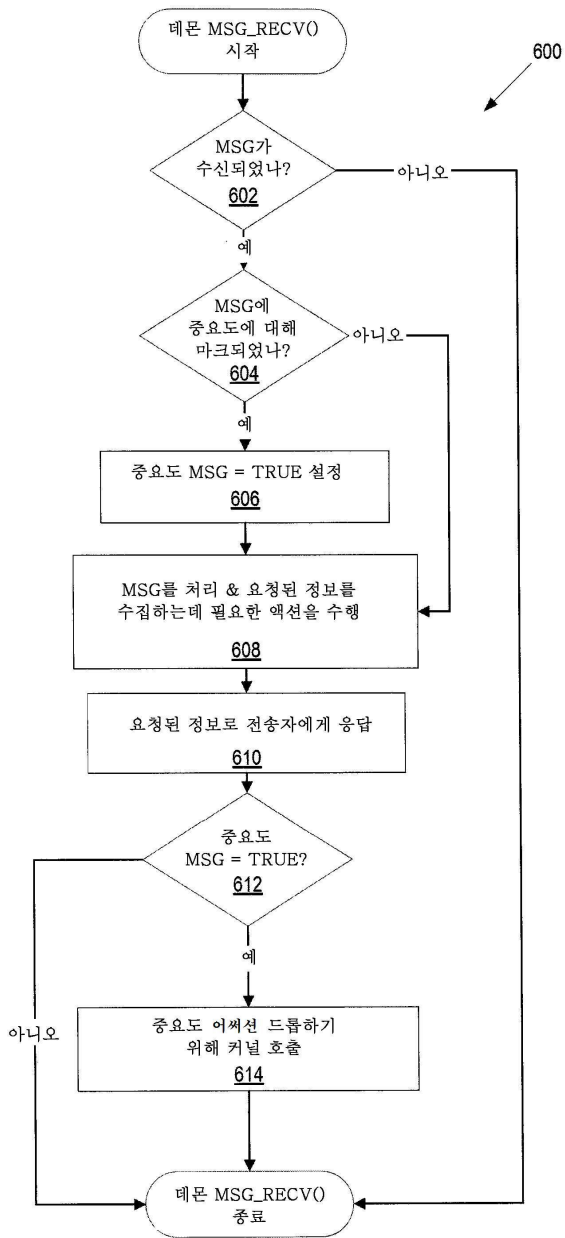
도면4



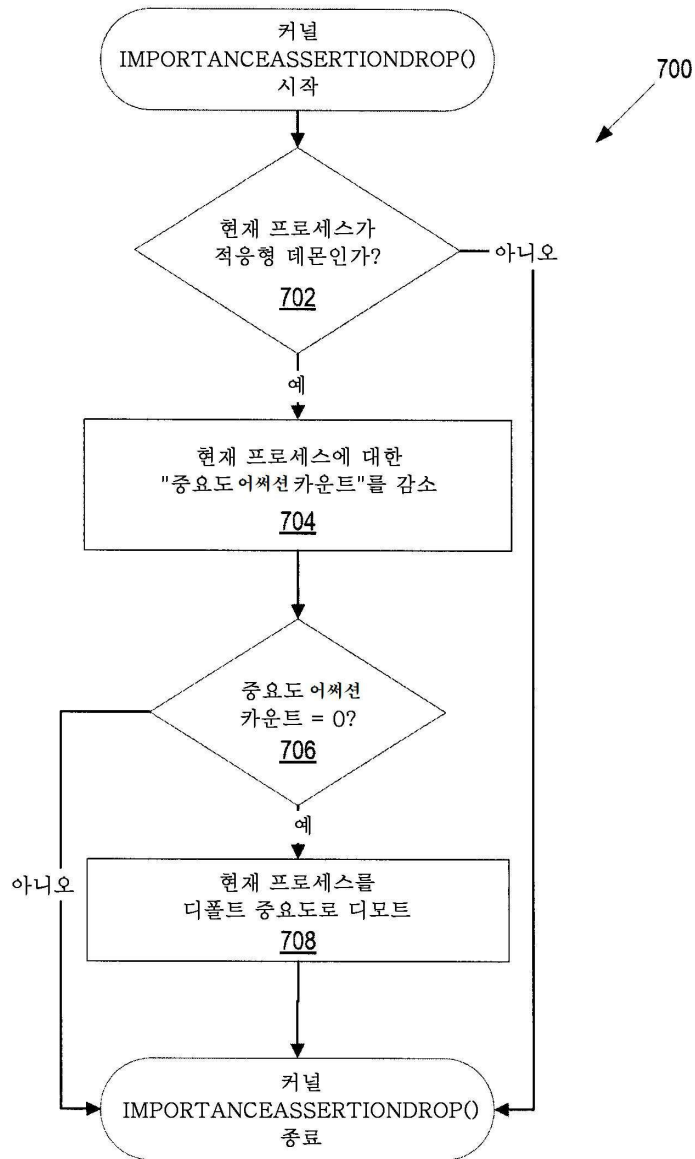
도면5



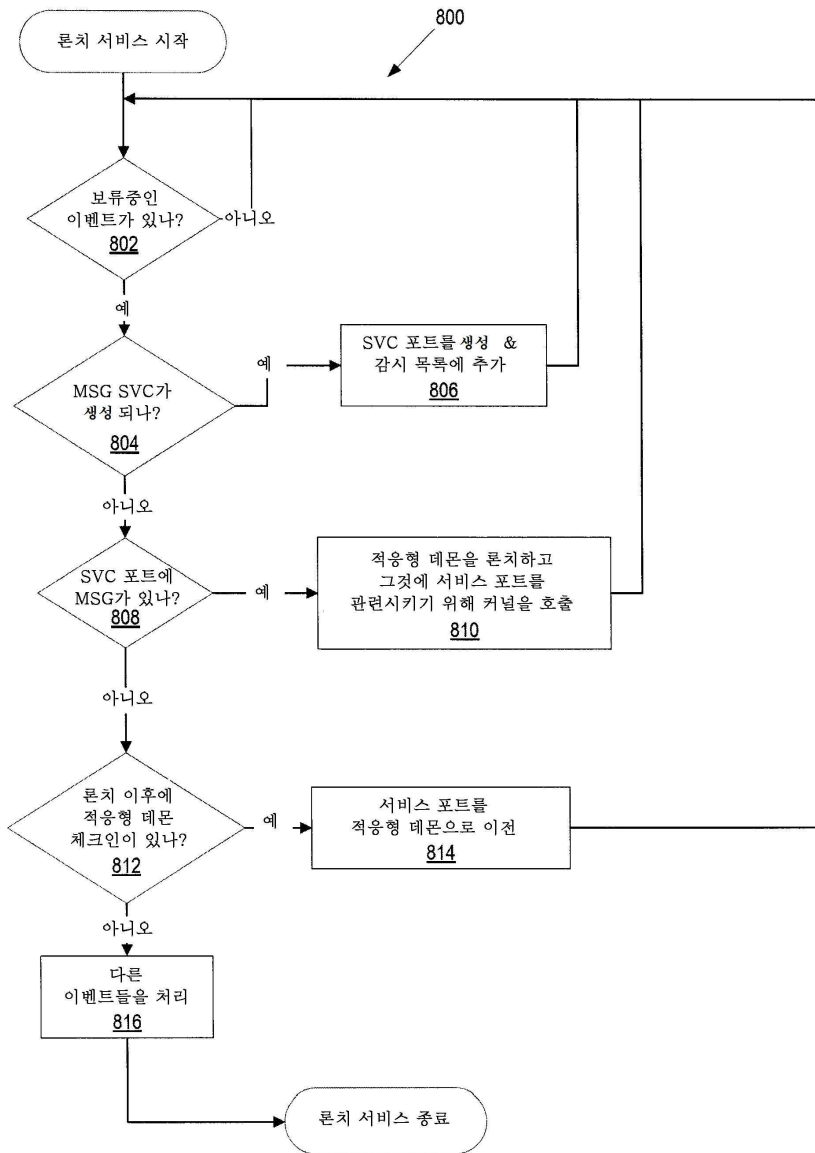
도면6



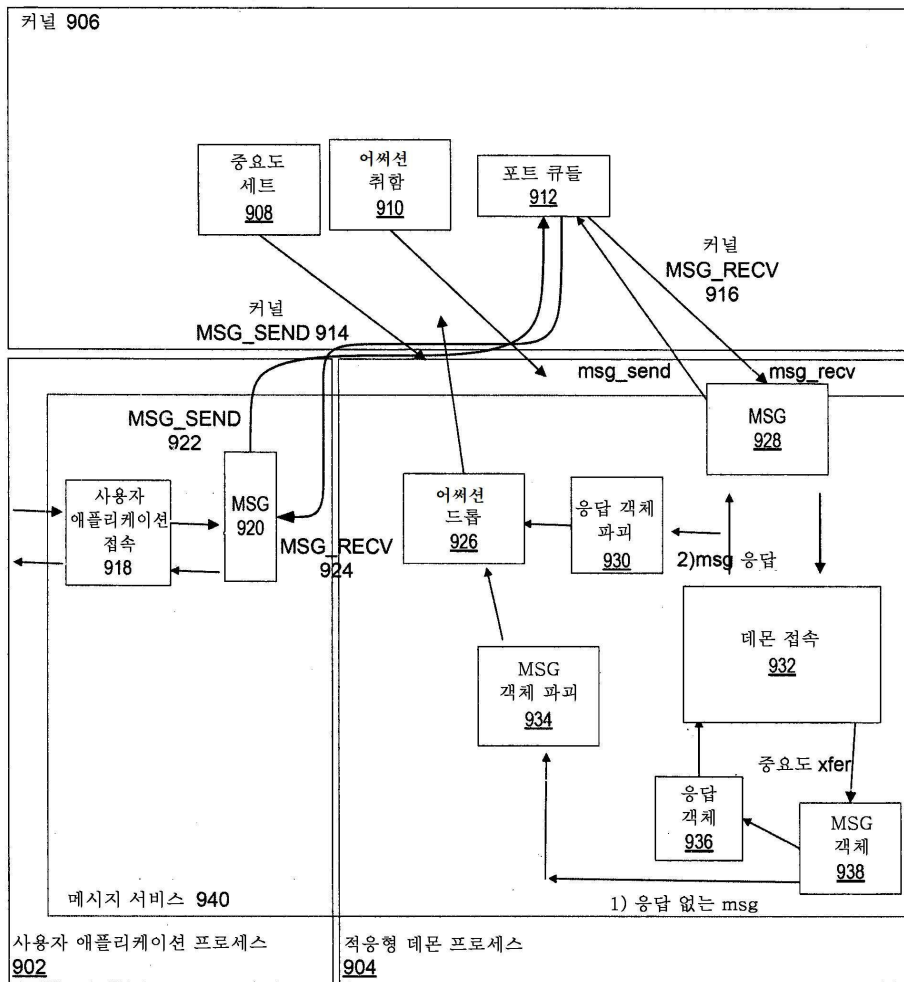
도면7



도면8

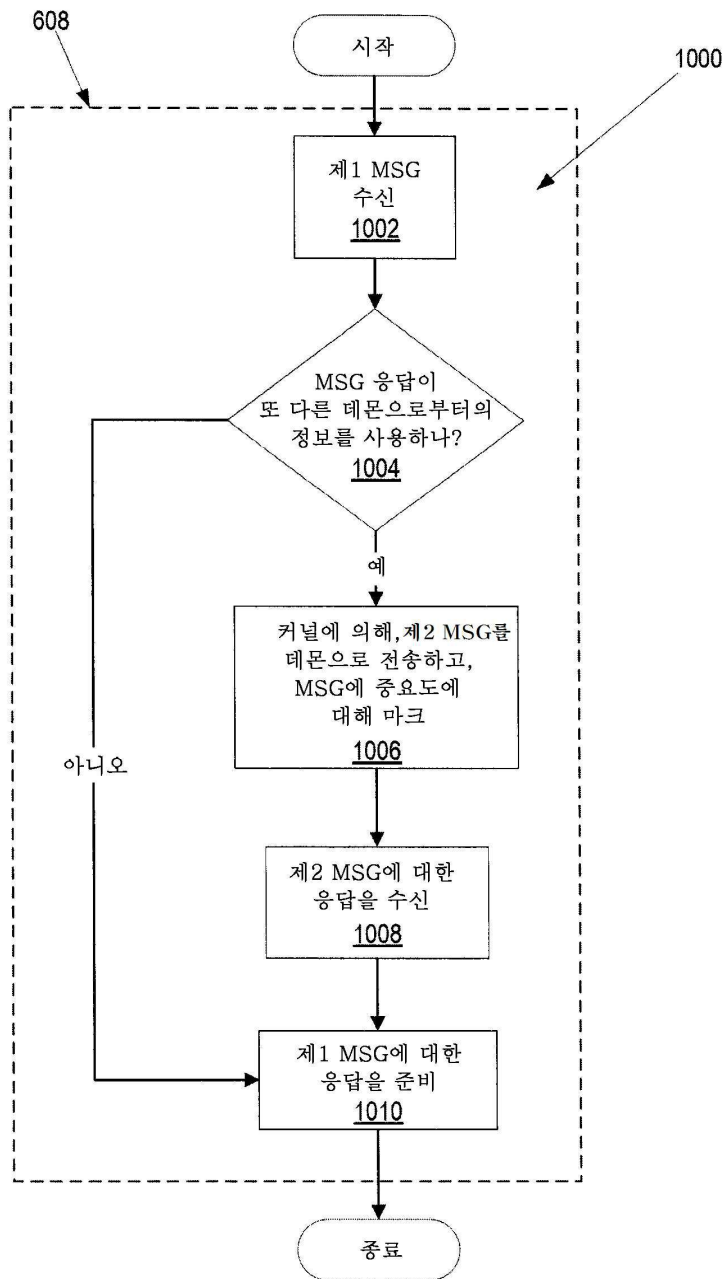


도면9

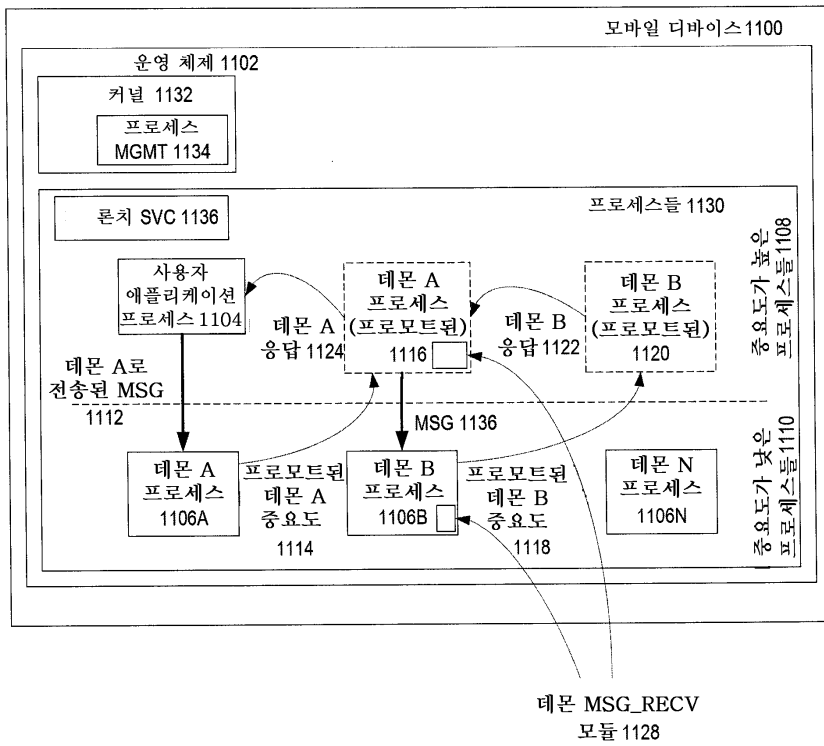


900

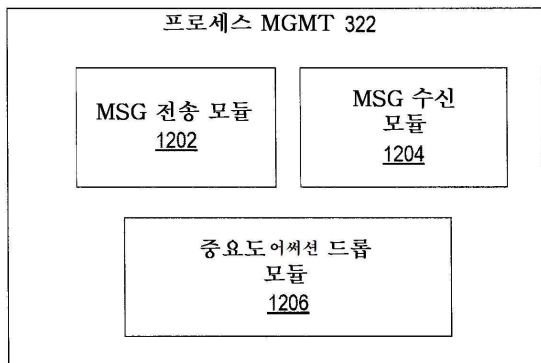
도면10



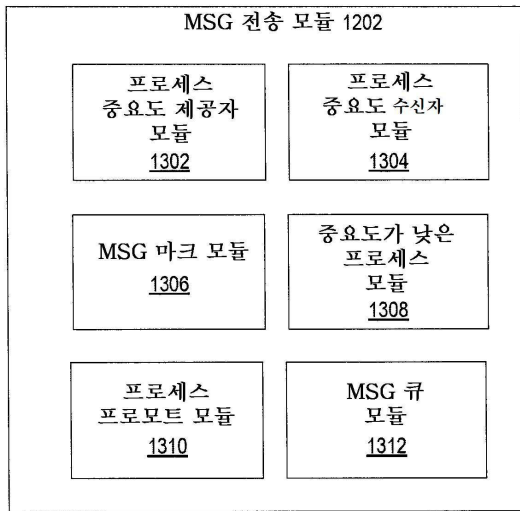
도면11



도면12



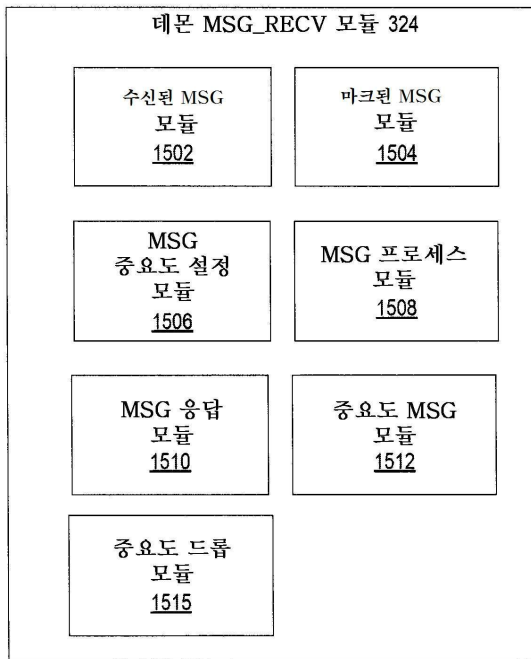
도면13



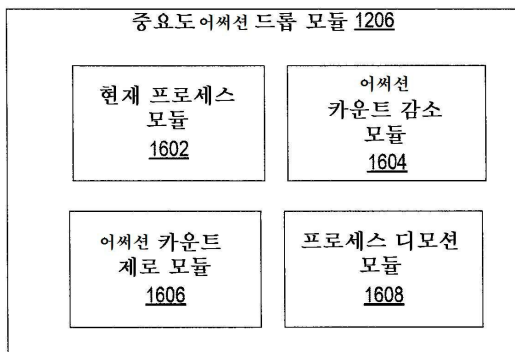
도면14



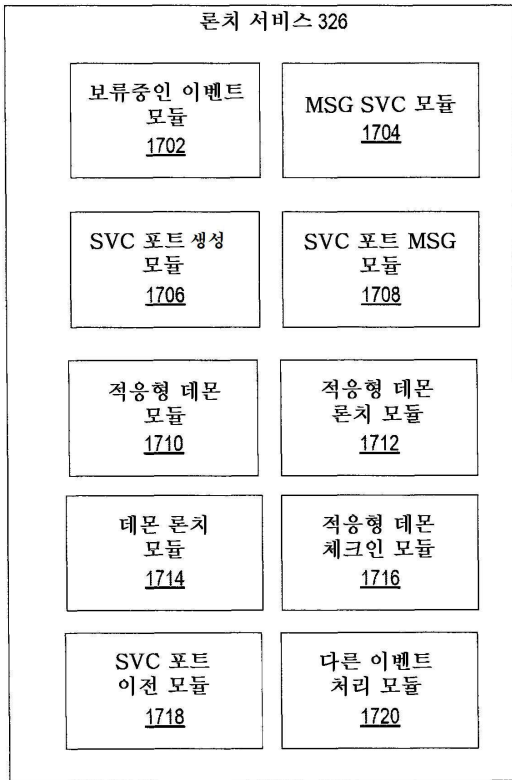
도면15



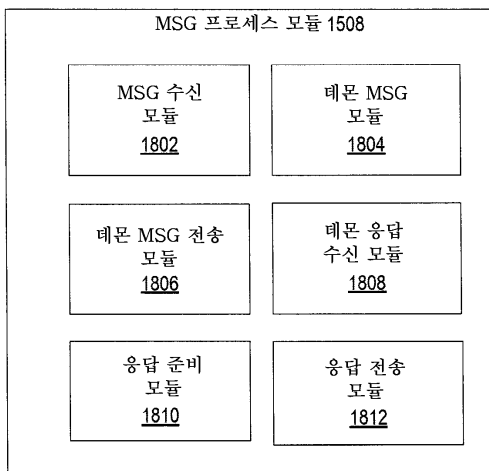
도면16



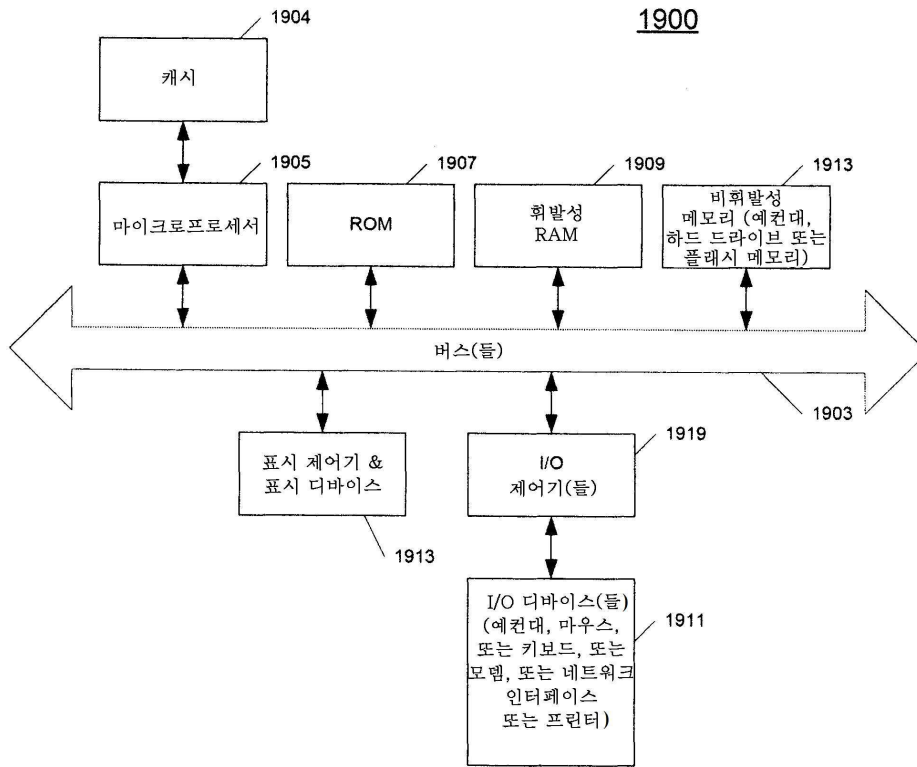
도면17



도면18



도면19



도면20

