



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월27일
(11) 등록번호 10-1963051
(24) 등록일자 2019년03월21일

(51) 국제특허분류(Int. Cl.)

G06F 9/44 (2018.01)

(52) CPC특허분류

G06F 9/44 (2013.01)

(21) 출원번호 10-2017-0166785

(22) 출원일자 2017년12월06일

심사청구일자 2017년12월06일

(56) 선행기술조사문헌

KR1020120072105 A

KR100862937 B1

JP2014526071 A

KR1020060041228 A

(73) 특허권자

고려대학교 산학협력단

서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)

(72) 발명자

김효곤

서울특별시 송파구 양재대로 1218, 214동 502호 (방이동, 올림픽선수기자촌아파트)

임승현

서울특별시 동대문구 전농로16길 51, 106동 201호 (전농동, 래미안아름숲)

(74) 대리인

송인호, 윤형근, 최영중, 최관락

전체 청구항 수 : 총 7 항

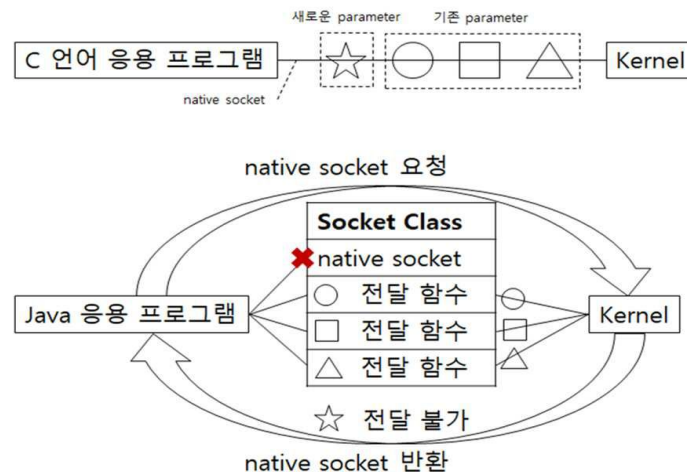
심사관 : 최정권

(54) 발명의 명칭 Java 기반의 응용 프로그램을 실행하는 단말 장치 및 이의 제어 방법

(57) 요약

Java 기반의 응용 프로그램을 실행하는 단말 장치 및 이의 제어 방법이 개시된다. 개시된 단말 장치는, 메모리; 및 상기 메모리와 연결되며, 운영 체제 상에서 Java 기반의 응용 프로그램을 실행하는 프로세서;를 포함하되, 상기 응용 프로그램 내의 클래스에는 복수의 네이티브 소켓 및 미리 정의된 복수의 함수가 포함되고, 상기 복수의 함수는 정형화된 매개 변수를 요청하기 위한 적어도 하나의 요청 함수 및 상기 정형화된 매개 변수의 값을 전달하는 적어도 하나의 전달 함수를 포함하며, 상기 응용 프로그램이 미리 정의된 특정 동작을 수행하는 경우, 상기 응용 프로그램의 커널은 상기 응용 프로그램으로 상기 복수의 네이티브 소켓 중 하나의 네이티브 소켓을 반환한다.

대표도 - 도4



명세서

청구범위

청구항 1

메모리; 및

상기 메모리와 연결되며, 운영 체제 상에서 자바 기반의 응용 프로그램을 실행하는 프로세서;를 포함하되,

상기 응용 프로그램 내의 클래스에는 복수의 네이티브 소켓 및 미리 정의된 복수의 함수가 포함되고, 상기 복수의 함수는 정형화된 매개 변수를 요청하기 위한 적어도 하나의 요청 함수 및 상기 정형화된 매개 변수의 값을 전달하는 적어도 하나의 전달 함수를 포함하며,

상기 응용 프로그램이 미리 정의된 특정 동작을 수행하는 경우, 상기 운영 체제의 커널은 상기 응용 프로그램으로 상기 복수의 네이티브 소켓 중 하나의 네이티브 소켓을 반환하는 것을 특징으로 하는 단말 장치.

청구항 2

제1항에 있어서,

상기 하나의 네이티브 소켓의 반환은 상기 단말 장치가 외부 장치와의 TCP 통신 연결이 수행되기 전에 수행되는 것을 특징으로 하는 단말 장치.

청구항 3

제1항에 있어서,

상기 응용 프로그램은 상기 하나의 네이티브 소켓을 통해 정형화되지 않은 매개 변수를 상기 커널로 전달하는 것을 특징으로 하는 단말 장치.

청구항 4

제3항에 있어서,

상기 특정 동작은, 상기 응용 프로그램이 상기 정형화된 매개 변수 중 어느 하나의 매개 변수를 요청하는 요청 함수를 상기 클래스에서 호출하여 어느 하나의 매개 변수를 획득한 후, 상기 어느 하나의 매개 변수와 대응되는 전달 함수를 상기 클래스에서 호출하여 상기 어느 하나의 매개 변수를 상기 커널로 전달한 후, 상기 요청 함수를 상기 클래스에서 재호출하는 동작이고,

상기 하나의 네이티브 소켓은 상기 복수의 네이티브 소켓 중 상기 어느 하나의 매개 변수와 대응되는 소켓인 것을 특징으로 하는 단말 장치.

청구항 5

제4항에 있어서,

상기 적어도 하나의 요청 함수는, getTCPNoDelay(), getSoLinger(), getSoTimeout(), getReceiveBufferSize(), getSendBufferSize(), getKeepAlive() 및 getReuseAddress()를 포함하는 것을 특징으로 하는 단말 장치.

청구항 6

제5항에 있어서,

상기 적어도 하나의 전달 함수는, setTcpNoDelay(), setSoLinger(), setSoTimeout(), setReceiveBufferSize(), setSendBufferSize(), setKeepAlive(), setReuseAddress()를 포함하는 것을 특징으로 하는 단말 장치.

청구항 7

운영 체제 상에서 자바 기반의 응용 프로그램을 실행하는 프로세서를 포함하는 단말 장치의 제어 방법에

있어서,

상기 응용 프로그램이 미리 정의된 특정 동작을 수행하는 단계; 및

상기 응용 프로그램이 상기 특정 동작을 수행하는 경우, 상기 운영 체제의 커널이 상기 응용 프로그램 내의 클래스에 포함된 복수의 네이티브 소켓 중 하나의 네이티브 소켓을 상기 응용 프로그램으로 반환하는 단계;를 포함하는 것을 특징으로 하는 단말 장치의 제어 방법.

발명의 설명

기술 분야

[0001] 본 발명의 실시예들은 Java 기반의 응용 프로그램에서 네이티브 소켓을 획득하는 동작을 수행하는 단말 장치 및 이의 제어 방법에 관한 것이다.

배경 기술

[0002] Java는 객체 지향 프로그래밍 언어로서 보안성이 뛰어나며 컴파일한 코드는 다른 운영 체제에서 사용할 수 있도록 클래스(class)로 제공된다. 객체 지향 언어인 C++ 언어의 객체 지향적인 장점을 살리면서 분산 환경을 지원하며 더욱 효율적이다.

[0003] 특히, Java에서는 TCP 통신을 위한 소켓(socket)을 네이티브 소켓(native socket)이 아닌 클래스의 형태로 제공하고, 네이티브 소켓을 사용하기 위한 함수를 제공한다. 소켓은 사용자 공간(user space)와 리눅스 등의 커널 공간(kernel space)를 이어주는 인터페이스이다.

[0004] 그러나, 기존에 정의된 함수를 사용하지 않고, 새로운 방식으로 사용하는 것은 Java의 TCP 소켓 통신에서는 쉽지 않다. 이에 따라, Java의 TCP 소켓 통신은 커널과의 정보 교환을 제한적으로 지원한다. 네이티브 소켓은 Java의 TCP 소켓 통신에서 커널과의 인터페이스이기 때문이다.

[0005] 한편, TCP는 전송 계층에서 신뢰성 있는 전송을 수행하기 위한 프로토콜로 연결 지향적이다. TCP를 통해 두 호스트 장치가 연결되며, 이러한 연결의 매개 변수(parameter)를 설정해주기 위해서 응용 프로그램에서는 TCP 소켓 옵션(socket option)을 사용한다. 그러나, 기존의 정형화된 매개 변수를 설정하는 것이 아니고 새로운 매개 변수를 설정하는 경우, Java에서는 새로운 TCP 소켓 옵션을 사용하기 어려운 단점이 있다.

발명의 내용

해결하려는 과제

[0006] 상기한 바와 같은 종래기술의 문제점을 해결하기 위해, 본 발명에서는 Java 기반의 응용 프로그램에서 네이티브 소켓을 획득하여 새로운 매개 변수를 설정하는 동작을 수행하는 단말 장치 및 이의 제어 방법을 제안하고자 한다.

[0007] 본 발명의 다른 목적들은 하기의 실시예를 통해 당업자에 의해 도출될 수 있을 것이다.

과제의 해결 수단

[0008] 상기한 목적을 달성하기 위해 본 발명의 바람직한 일 실시예에 따르면, 메모리; 및 상기 메모리와 연결되며, 운영 체제 상에서 Java 기반의 응용 프로그램을 실행하는 프로세서;를 포함하되, 상기 응용 프로그램 내의 클래스에는 복수의 네이티브 소켓 및 미리 정의된 복수의 함수가 포함되고, 상기 복수의 함수는 정형화된 매개 변수를 요청하기 위한 적어도 하나의 요청 함수 및 상기 정형화된 매개 변수의 값을 전달하는 적어도 하나의 전달 함수를 포함하며, 상기 응용 프로그램이 미리 정의된 특정 동작을 수행하는 경우, 상기 응용 프로그램의 커널은 상기 응용 프로그램으로 상기 복수의 네이티브 소켓 중 하나의 네이티브 소켓을 반환하는 것을 특징으로 하는 단말 장치가 제공된다.

[0009] 상기 하나의 네이티브 소켓의 반환은 상기 단말 장치가 상기 외부 장치와의 TCP 통신 연결이 수행되기 전에 수행될 수 있다.

[0010] 상기 응용 프로그램은 상기 하나의 네이티브 소켓을 통해 정형화되지 않은 매개 변수를 상기 커널로 전달할 수 있다.

[0011] 상기 특정 동작은, 상기 응용 프로그램이 상기 정형화된 매개 변수 중 어느 하나의 매개 변수를 요청하는 요청 함수를 상기 클래스에서 호출하여 상기 어느 하나의 매개 변수를 획득한 후, 상기 어느 하나의 매개 변수와 대응되는 전달 함수를 상기 클래스에서 호출하여 상기 어느 하나의 매개 변수를 상기 커널로 전달한 후, 상기 요청 함수를 상기 클래스에서 재호출하는 동작이고, 상기 하나의 네이티브 소켓은 상기 복수의 네이티브 소켓 중 상기 어느 하나의 매개 변수와 대응되는 소켓일 수 있다.

[0012] 상기 적어도 하나의 요청 함수는, getTCPNoDelay(), getSoLinger(), getSoTimeout(), getReceiveBufferSize(), getSendBufferSize(), getKeepAlive() 및 getReuseAddress()를 포함할 수 있다.

[0013] 상기 적어도 하나의 전달 함수는, setTcpNoDelay(), setSoLinger(), setSoTimeout(), setReceiveBufferSize(), setSendBufferSize(), setKeepAlive(), setReuseAddress()를 포함할 수 있다.

[0014] 또한, 본 발명의 다른 실시예에 따르면, 운영 체제 상에서 Java 기반의 응용 프로그램을 실행하는 프로세서를 포함하는 단말 장치의 제어 방법에 있어서, 상기 응용 프로그램이 미리 정의된 특정 동작을 수행하는 단계; 및 상기 응용 프로그램이 상기 특정 동작을 수행하는 경우, 상기 운영 체제의 커널이 상기 응용 프로그램 내의 클래스에 포함된 복수의 네이티브 소켓 중 하나의 네이티브 소켓을 상기 응용 프로그램으로 반환하는 단계;를 포함하는 것을 특징으로 하는 단말 장치의 제어 방법이 제공된다.

발명의 효과

[0015] 본 발명에 따르면, Java 기반의 응용 프로그램에서 네이티브 소켓을 획득하여 새로운 매개 변수를 설정하는 동작을 수행할 수 있다.

[0016] 또한, 본 발명의 효과는 상기한 효과로 한정되는 것은 아니며, 본 발명의 상세한 설명 또는 특허청구범위에 기재된 발명의 구성으로부터 추론 가능한 모든 효과를 포함하는 것으로 이해되어야 한다.

도면의 간단한 설명

[0017] 도 1은 본 발명의 일 실시예에 따른 단말 장치의 개략적인 구성을 도시한 도면이다.

도 2 및 도 3은 본 발명의 일 실시예에 따른 단말 장치의 제어 방법의 흐름도를 도시한 도면이다.

도 4는 본 발명의 개념을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0018] 본 명세서에서 사용되는 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "구성된다" 또는 "포함한다" 등의 용어는 명세서상에 기재된 여러 구성 요소들, 또는 여러 단계들을 반드시 모두 포함하는 것으로 해석되지 않아야 하며, 그 중 일부 구성 요소들 또는 일부 단계들은 포함되지 않을 수도 있고, 또는 추가적인 구성 요소 또는 단계들을 더 포함할 수 있는 것으로 해석되어야 한다. 또한, 명세서에 기재된 "...부", "모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어 또는 소프트웨어로 구현되거나 하드웨어와 소프트웨어의 결합으로 구현될 수 있다.

[0019] 이하, 본 발명의 다양한 실시예들을 첨부된 도면을 참조하여 상술한다.

[0020] 도 1은 본 발명의 일 실시예에 따른 단말 장치(100)의 개략적인 구성을 도시한 도면이다.

[0021] 도 1을 참조하면, 단말 장치(100)는 Java 기반의 응용 프로그램이 실행되는 호스트 장치일 수 있으며, 복수의 통신 포트(110), 메모리(120) 및 프로세서(130)를 포함한다. 이하, 각 구성 요소 별로 그 기능을 상세하게 설명하기로 한다.

[0022] 복수의 통신 포트(110) 각각은 적어도 하나의 외부 장치와 통신하기 위한 기능을 수행한다. 이 때, 통신 포트(110)는 TCP 또는 UDP 방식으로 통신을 수행할 수 있다.

[0023] 메모리(120)는 휘발성 및/또는 비휘발성 메모리일 수 있고, 단말 장치(100)의 적어도 하나의 다른 구성요소에 관계된 명령어 또는 데이터를 저장한다. 특히, 메모리(120)는 운영 체제(OS) 및 운영 체제 상에서 실행되는 복수의 Java 기반의 응용 프로그램과 관련된 명령 또는 데이터를 저장할 수 있다.

[0024] 프로세서(130)는 중앙처리장치, 애플리케이션 프로세서, 또는 커뮤니케이션 프로세서 중 하나 또는 그 이상을

포함할 수 있다. 예를 들면, 프로세서(130)는 단말 장치(100)의 적어도 하나의 다른 구성요소들의 제어 및/또는 통신에 관한 연산이나 데이터 처리를 실행할 수 있다. 특히, 프로세서(130)는 Java 기반 응용 프로그램을 실행할 수 있으며, 이는 미리 설치된 운영 체제 상에서 실행될 수 있다.

[0025] 이하, 도 2 및 도 3을 참조하며 상기에서 설명한 구조를 가지는 단말 장치(100)의 동작을 상세하게 설명하기로 한다.

[0026] 도 2 및 도 3은 본 발명의 일 실시예에 따른 단말 장치(100)의 제어 방법의 흐름도를 도시한 도면이다. 이 때, 상기 방법은 프로세서(130)를 중심으로 수행될 수 있으며, 단말 장치(100)와 외부 장치와의 TCP 통신이 연결되기 전에 수행될 수 있다.

[0027] 이하, 각 단계 별로 수행되는 과정을 상세하게 설명하기로 한다.

[0028] 도 2를 참조하면, 단계(202)에서, 프로세서(130)에서 실행되는 Java 기반 응용 프로그램은 미리 정의된 특정 동작을 수행한다.

[0029] 보다 상세하게, Java 기반 응용 프로그램 내의 클래스(class)에는 복수의 네이티브 소켓(native socket) 및 미리 정의된 복수의 함수(function) 또는 메소드(method)가 포함된다. 여기서, 복수의 함수는 정형화된 매개 변수(즉, 미리 정의된 매개 변수)를 요청하기 위한 적어도 하나의 요청 함수 및 정형화된 매개 변수의 값을 전달하는 적어도 하나의 전달 함수를 포함할 수 있다.

[0030] 그리고, 도 3에서는 상기 단계(202)의 상세한 단계, 즉 상기한 특정 동작의 상세한 구성을 도시하고 있다.

[0031] 도 3을 참조하면, 단계(302)에서, Java 기반 응용 프로그램은 정형화된 매개 변수 중 어느 하나의 매개 변수를 요청하는 요청 함수를 클래스에서 호출한다. 이에 따라, 상기한 어느 하나의 매개 변수가 획득된다.

[0032] 본 발명의 일 실시예에 따르면, 요청 함수는 `getTCPNoDelay()`, `getSoLinger()`, `getSoTimeout()`, `getReceiveBufferSize()`, `getSendBufferSize()`, `getKeepAlive()` 및 `getReuseAddress()` 중 어느 하나의 함수일 수 있다. 그러나, 본 발명이 이에 한정되는 것은 아니며 클래스에 포함되는 모든 요청 함수가 사용될 수 있다.

[0033] 그리고, 단계(304)에서, Java 기반 응용 프로그램은 상기한 어느 하나의 매개 변수와 대응되는 전달 함수를 클래스에서 호출한다. 그리고, 단계(306)에서, Java 기반 응용 프로그램은 상기한 어느 하나의 매개 변수를 커널로 전달한다.

[0034] 본 발명의 일 실시예에 따르면, 전달 함수는 `setTcpNoDelay()`, `setSoLinger()`, `setSoTimeout()`, `setReceiveBufferSize()`, `setSendBufferSize()`, `setKeepAlive()`, `setReuseAddress()` 중 어느 하나의 함수일 수 있다. 그러나, 본 발명이 이에 한정되는 것은 아니며 클래스에 포함되는 모든 전달 함수가 사용될 수 있다.

[0035] 그 후, 단계(308)에서, Java 기반 응용 프로그램은 단계(302)와 동일한 동작을 수행한다. 즉, 단계(308)에서, Java 기반 응용 프로그램은 상기한 어느 하나의 매개 변수를 요청하는 요청 함수를 클래스에서 호출한다.

[0036] 다시, 도 2를 참조하면, Java 기반 응용 프로그램이 미리 정의된 특정 동작을 수행하는 경우, 단계(204)에서, 커널은 Java 기반 응용 프로그램으로 복수의 네이티브 소켓 중 하나의 네이티브 소켓을 반환한다.

[0037] 본 발명의 일 실시예에 따르면, 하나의 네이티브 소켓은 상기한 어느 하나의 매개 변수와 관련된 네이티브 소켓일 수 있으며, 커널은 하나의 네이티브 소켓의 정보인 정수값을 반환할 수 있다.

[0038] 즉, 도 3에서 설명한 특정 동작(패턴)은 Java 기반 응용 프로그램에서 일반적으로 수행되지 않는 무의미한 패턴 동작을 의미하는 것으로서, Java 기반 응용 프로그램에서 상기한 어느 하나의 매개 변수와 관련된 특정 동작(패턴)이 수행되면, 커널은 상기한 어느 하나의 매개 변수에 속하는 네이티브 소켓의 정보인 정수값을 가지고 있다가, Java 기반 응용 프로그램에서 시스템 콜(system call)을 호출하면 커널은 해당 네이티브 소켓의 정보인 정수값을 반환한다.

[0039] 마지막으로, 단계(206)에서, Java 기반 응용 프로그램은 하나의 네이티브 소켓을 통해 정형화되지 않은 매개 변수를 커널로 전달한다. 즉, Java 기반 응용 프로그램은 상기한 어느 하나의 매개 변수와 관련된 네이티브 소켓을 통해 정형화되지 않은 새로운 매개 변수를 커널로 전달할 수 있다.

[0040] 이하, 도 4를 참조하여 본 발명의 개념을 설명하면 다음과 같다.

[0041] C 언어 기반의 응용 프로그램의 경우, TCP 연결 시 클래스에 속하지 않는 네이티브 소켓을 사용하기 때문에 정

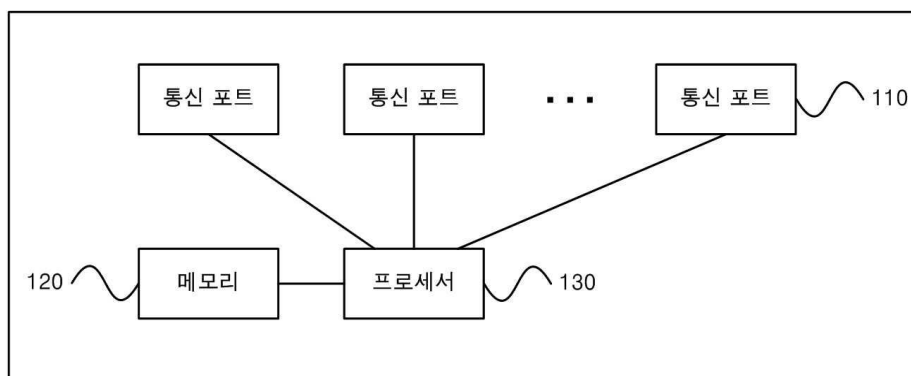
형화되지 않은 새로운 매개 변수를 설정하기가 용이하다.

- [0042] 하지만, Java 기반의 응용 프로그램의 경우, 클래스 내부에 네이티브 소켓을 넣어두고, 기존에 정해진 함수들만을 통해서 네이티브 소켓에 접근하도록 한다. 따라서, 정형화되지 않은 새로운 매개 변수를 설정하기 위해서는 새로운 함수를 정의해야 하나, 이는 Java 자체를 변경하는 일이기 때문에 거의 불가능하다. 특히, TCP 연결 전에 새로운 매개 변수를 설정하는 것은 불가능하다.
- [0043] 예를 들어, 빅 데이터 처리를 위해 사용되는 Java 기반의 소프트웨어 프레임워크인 하둡의 경우, 데이터를 저장하기 위해 TCP 통신을 사용하며, 이들의 성능을 높이기 위해 TCP 통신 연결 전에 새로운 매개 변수를 설정해주어야 하는 상황이라 가정하자. 일례로, 하둡에서 데이터를 데이터 노드에 저장하는 경우, 새로운 매개 변수는 데이터 노드의 IP 주소 및 TCP 포트 주소 등의 정보를 TCP SYN 세그먼트의 TCP 소켓 옵션을 통해 전달하고자 하는 경우 필요한 매개 변수이다.
- [0044] 종래의 경우, 이미 만들어진 하둡을 새롭게 만들거나, TCP 통신 연결을 맺는 부분만을 새롭게 만들어 기존 Java와 인터페이스를 매칭하는 작업이 수행되어야 했다. 그러나, 이러한 작업은 상당한 시간이 소요되며 하위 호환성(backward compatibility) 문제를 야기할 수 있다.
- [0045] 그러나, 본 발명은 상기에서 설명한 방법을 통해 네이티브 소켓을 전달받아 정형화되지 않은 새로운 매개 변수를 Java 기반 응용 프로그램에서 커널로 전달할 있는 장점이 있다.
- [0046] 또한, 본 발명의 실시예들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 일 실시예들의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.
- [0047] 이상과 같이 본 발명에서는 구체적인 구성 요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나 이는 본 발명의 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 본 발명이 속하는 분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명의 사상은 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐 아니라 이 특허청구범위와 균등하거나 등가적 변형이 있는 모든 것들은 본 발명 사상의 범주에 속한다고 할 것이다.

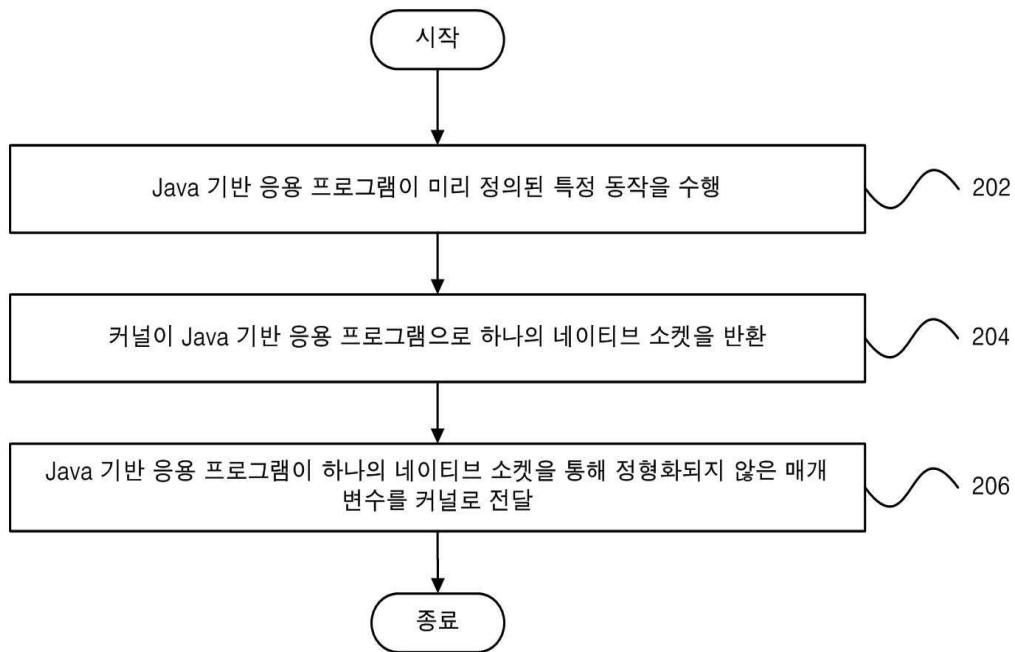
도면

도면1

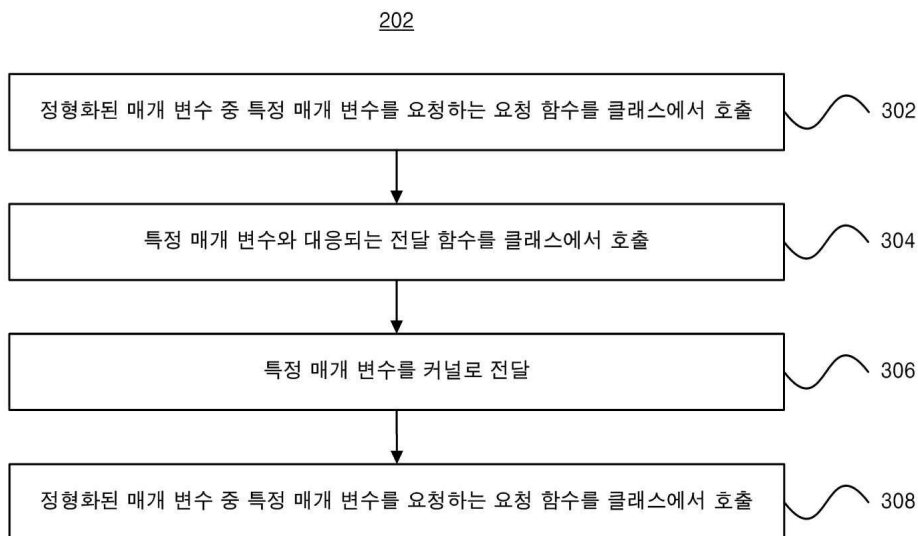
100



도면2



도면3



도면4

