WHAT IS CLAIMED IS:

1. An apparatus for managing digital rights, the apparatus comprising:

an DRM agent unit managing application programs to which DRM (Digital Rights Management) is to be applied and a right to contents processed by the application programs to which DRM is to be applied and encrypting or decrypting the content processed by the application programs to which DRM is to be applied;

a DRM rights management unit authenticating a user and managing a user right to the contents processed by the application programs to which DRM is to be applied; and

a kernel API (Application Program Interface) hooking unit monitoring input/output of a file through hooking of kernel native APIs, requesting the DRM rights management unit to verify a user right to contents to be processed when the contents are detected as being processed by the application programs to which DRM is to be applied, and requesting the DRM agent unit to encrypt or decrypt the contents to be processed when the user right to the contents to be processed is verified,

wherein the DRM agent unit and the DRM rights management unit operate at user level, and the kernel API hooking unit operates at kernel level,

wherein the DRM agent unit, the DRM rights management unit, and the kernel API hooking unit are driven for each application program to which DRM is to be applied.

2. The apparatus of claim 1, wherein the DRM agent unit comprises:

a user policy management unit changing rights of a user or rights to contents on the basis of rights information received from the user or a management server managing DRM policies;

a document rights management unit managing a list of application programs to which DRM is to be applied and managing rights information for each document and each application program to which DRM is to be applied;

an injection unit monitoring application programs running on an operating system and driving the DRM agent unit, the DRM rights management unit, and the kernel API hooking unit for each application program when the application program to which DRM is to be applied is running; and

an encryption/decryption unit encrypting or decrypting the contents and returning

the encrypted or decrypted contents to the kernel API hooking unit in response to the request for encryption or decryption of the contents from the kernel API hooking unit.

3. The apparatus of claim 2, wherein the rights information received from the management server comprises rights information for each content or application program to which DRM is to be applied, information on the application programs to which DRM is to be applied, and information on allowance/limitation/change of the user right.

4. The apparatus of claim 2, wherein, when an application program is detected as being running, the injection unit inquires of the document rights management unit whether the running application program is the application program to which DRM is to be applied.

5. The apparatus of claim 1, wherein the DRM rights management unit comprises:

a user authentication unit verifying the user right to the contents or the application programs to which DRM is to be applied and outputting an authentication result on the user; and

a user rights management unit managing the user rights to read, write, modification, capture, clipboard use, and print on the basis of the authentication result on the user and returning a verification result in response to a request for verifying of the right to the contents to be processed from the kernel API hooking unit.

6. The apparatus of claim 5, wherein the DRM rights management unit further comprises a use history management unit managing a use history including opening, modification, and storage of the contents to which DRM is applied.

7. The apparatus of claim 1, wherein the kernel API hooking unit comprises:

a rights control unit requesting the DRM agent unit and the DRM rights management unit to verify the user right to the contents corresponding to file input/output events and outputting the verification result; and

an encryption/decryption control unit hooking the kernel native APIs to detect the file input/output events, requesting the DRM rights management unit to verify the user

right to the contents to be processed corresponding to the file input/output events, and delivering the contents to be processed to the DRM agent unit to direct encryption or decryption on the basis of the verification result.

8. ~~The apparatus of claim 7, wherein the kernel API hooking unit further comprises a cache processing unit receiving previously encrypted or decrypted contents from the encryption/decryption control unit and storing the contents in a built-in cache memory, thereby preventing an opened document from being encrypted or decrypted repeatedly, and when an inquiry whether the contents to be processed have been encrypted or decrypted is input from the encryption/decryption control unit, searching the cache memory and returning the contents or outputting a notification that the contents has been not decrypted.~~

~~9.~~ The apparatus of claim 1, wherein the kernel API hooking unit comprises:

a rights control unit hooking the kernel native APIs to detect file input/output events, requesting the DRM agent unit and the DRM rights management unit to verify the user right to the contents to be processed corresponding to the detected file input/output events, and outputting the verification result; and

an encryption/decryption control unit delivering the contents to be processed to the DRM agent unit on the basis of the verification result to direct encryption or decryption.

9~~10~~. The apparatus of claim ~~9~~7 or 8, wherein the kernel API hooking unit further comprises a cache memory and stores ~~processing unit receiving previously encrypted or decrypted contents from the encryption/decryption control unit and storing the~~previously encrypted or decrypted contents delivered from the encryption/decryption control unit in ~~a built in~~the cache memory in order to~~, thereby~~ prevent~~ing~~ an opened document from being encrypted or decrypted repeatedly, and when an inquiry whether the contents to be processed have been encrypted or decrypted is input from the encryption/decryption control unit, searches~~ing~~ the cache memory and returning the contents or outputting a notification that the contents has been not decrypted.

10~~1~~. A method for managing digital rights performed in an apparatus for managing digital rights, the method comprising:

(a) injecting a DRM (Digital Rights Management) engine module into an application program to which DRM is to be applied when the application program to which DRM is to be applied is driven;

(b) monitoring input/output of a file through hooking a kernel native API (Application Program Interface), and when a content is detected as being processed by the application program to which DRM is to be applied, requesting for verifying a user right to the content to be processed in the application program to which DRM is to be applied;

(c) selectively encrypting or decrypting the content to be processed when the user right to the content to be processed is verified; and

(d) returning the encrypted or decrypted content to the kernel native API,

wherein step (a) and step (c) are performed at user level, and step (b) and step (d) are performed at kernel level.

112. The method of claim 101, wherein step (a) comprises injecting the DRM engine module into the detected application program, when the detected application program is in a predetermined list of application programs to which DRM is to be applied.

123. The method of claim 101, wherein the ~~right of the~~ user ~~or the~~ right set to the user or the content is changeable on the basis of rights information received from the user or a management server managing DRM policies.

134. The method of claim 123, wherein the rights information received from the management server comprises at least one information out of rights information for each content, ~~or~~rights information for each application program to which DRM is to be applied, information on the application program to which DRM is to be applied, and information on allowance/limitation/change of the user right.

145. The method of claim 101, further comprising (e) updating a use history which includes opening, modification, and storage of the content to which DRM is applied.

156. The method of claim 101, wherein step (c) comprises:

(c1) hooking the kernel native API to detect a file input/output event;

(c2) verifying the user right to the content to be processed corresponding to the file

input/output event, and requesting a rights management module operated at user level to verifyauthenticating the user, and manageing the user right to the content to be processed in the application program to which DRM is to be applied; and

(c3) delivering encrypting or decrypting the content to be processed to an encryption/decryption module operated at user level on the basis of the verification result by the rights management module and instructing to perform encryption or decryption.

167. The method of claim 156, wherein step (c) further comprises (c4) storing previously encrypted or decrypted contents in a cache memory in order to prevent an opened document from being encrypted or decrypted repeatedly, searching the cache memory when encryption or decryption is performed to the contents to be processed, and reading out the contents corresponding to the contents to be processed from the cache memory instead of performing step (c3)searching a cache memory which stores previously encrypted or decrypted contents and reading out the content, thereby preventing the content from being encrypted or decrypted repeatedly.

178. A program for executing the contents transmitting method according to any one of claims 10 to 16 in a computer.A non-transitory computer readable medium storing a program for executing the method for managing digital rights according to claim 11 in a computer.

18. A computer-readable recording medium storing the program according to claim 17.