SYSTEM AND METHOD FOR WRITING TO A DRIVE WHEN AN APPLICATION LACKS ADMINISTRATOR PRIVILEGES

BACKGROUND

[0001] To write (also referred as "print") to a disc in a CD drive, an interface is needed to allow an imaging application to send data to the drive. In a personal computer ("PC") with a Windows operating system ("OS"), access to the CD drive is usually done through the cdrom.sys class driver. Interfacing with the class driver requires very low level programming and is not an easy way for an imaging application to send to send print data.

[0002] As shown in Figures 1A and 1B, a standard computer system 100 includes a CD-ROM drive 108 which is modified to support printing and is programmed to recognize specific small computer system interface/attachment packet interface ("SCSI/ATAPI") commands that are only used for printing. In this system 100, a Windows application 102 (shown as "102A" and "102B") interfaces with an OS class driver 104 using input/output control ("IOCTL") language. However, because these commands are non-standard, it is necessary to bypass error checking protocols. Accordingly, the traditional approach is to use an SCSI-pass-through-interface ("SPTI") 106 interfaced with the OS driver 104.

[0003] As shown in Figure 1A, if the Windows application 102 is an application with administrator privileges 102A (i.e., the user has administrator privileges and this status is recognized by the system 100), when using an SPTI 106, the class driver 104 passes the raw print command data to the CD drive 108 via a port driver 130, an intelligent drive electronics (also known as "integrated drive electronics" or "IDE") bus driver 140, and an IDE bus 110; the SPTI does not interfere with or check the data. The driver 104 confirms the administrator privileges status before sending signals to the SPTI. For those applications with

administrator privileges 102A, the SPTI will send SCSI/ATAPI print commands to the CD drive 108. By way of contrast, as shown in Figure 1B, if the application does not have administrator privileges 102B (i.e., the user does not have administrator privileges and this status is recognized by the system 100), this status information is also passed to the driver 104; as a result, the driver 104 will not send print signals to the SPTI which, in turn, prevents the signals from reaching the CD drive 108.

[0004] As a result, to use an SPTI, an application must have administrator privileges (i.e., the user must have administrator privileges). However, as most PC users, particularly those on a network, do not have administrator privileges, the SPTI is usually unavailable, thereby preventing print commands from reaching the CD drive 108.

[0005] There are two predominant prior solutions to this administrator access problem. First, as shown in Figure 2, another system 150 has been developed using an IDE bus 110 but not using standard SCSI commands or drivers. This system 150 uses a second, customized driver 112 to control the data and control lines of the IDE bus 110 to send print data to the CD drive 108.

[0006] Unfortunately, as a result of using specialized commands to overcome the print access problem associated with no administrator privileges, this system 150 does not support standard drivers such as the class driver 104 that is already a part of the OS. As a result, the second driver 112 is necessary to act as an interface between the application without administrator privileges 102B and the IDE bus 110 via the port driver 130 and IDE bus driver 140. The second driver 112 may receive the same print commands as the OS driver 104 or may receive specialized print commands as a result of being interfaced with the custom driver. However, the second driver 112 is customized to convert the driver commands of the application 102B into non-SCSI commands which are sent through the bus 110 to the CD drive 108.

[0007] The second solution, shown in Figure 3, involves a system 175 which, like the standard OS shown in Figure 1A, uses SCSI commands to control the CD drive 108. However, in this system 175, the SCSI commands reach the CD

drive 108 via a third party CD driver 114 that is not part of the OS. An example of such a driver is the Advanced SCSI Programming Interface ("ASPI"). Unfortunately, third party drivers require license fees and have limited technical support in current operating systems. It should be recognized that in this system, the application without administrator privileges 102B interfaces with the system driver 104, the SPTI 106, and the third party driver 114. The OS driver 104 and the third party driver 114 receive the same print commands and unlike the previously described systems (in which administrator status signals were communicated to the system driver 104), in this system 175, the status signals are communicated to the SPTI 106. However, like the previously described system 150, the SPTI 106 is bypassed; in this system 175, the bypass is via the third party driver 114.

SUMMARY

[0008] One embodiment of the invention is a system and method which enable a user of a computer application without administrator privileges to write to a drive without using a third party driver and without using a customized driver acting in conjunction with an IDE bus. The system includes a driver programming interface in communication with the application, a service proxy in communication with the driver interface, a service with administrator privileges in communication with the service proxy, a driver in communication with the service with administrator privileges, and a drive in communication with the driver.

[0009] These and other features, aspects, and advantages of the present invention will become more apparent from the following description, appended claims, and accompanying exemplary embodiments shown in the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] Figure 1A is a schematic diagram of a standard operating system in which an application is being run with administrator privileges and is able to send write commands to a drive;

[0011] Figure 1B is a schematic diagram of a standard operating system in which an application is being run without administrator privileges and is unable to send write commands to a drive;

[0012] Figure 2 is a schematic diagram of a first prior art system enabling an application without administrator privileges to write to a drive;

[0013] Figure 3 is a schematic diagram of a second prior art system enabling an application without administrator privileges to write to a drive; and

[0014] Figure 4 is a schematic diagram of one embodiment of a system which enables an application without administrator privileges to write to a drive without using a second, customized driver working in conjunction with a IDE bus and without using a third party driver.

DETAILED DESCRIPTION

[0015] One embodiment of the invention comprises a system for sending write commands to a drive from a computer application lacking administrator privileges. This system includes: (a) a computer application lacking administrator privileges; (b) a driver programming interface in communication with the computer application; (c) a service proxy in communication with the driver programming interface; (d) a service with administrator privileges in communication with the service proxy; (e) a driver in communication with the service with administrator privileges, wherein the driver is either in communication with an SPTI or is configured in SPTI mode; and (f) a drive in communication with the SPTI or with the SPTI configured driver.

[0016] Another embodiment of the invention comprises a method of sending write commands to a drive from a computer application which does not have administrator privileges. This method includes: (a) transmitting write commands between a computer application which does not have administrator privileges and a driver programming interface; (b) transmitting second commands based at least in part on the write commands between the driver programming interface and a service proxy; (c) transmitting third commands based at least in part on the second commands between the service proxy and a service with administrator privileges; (d) transmitting fourth commands based

at least in part on the third commands between the service with administrator privileges and a driver; (e) transmitting fifth commands based at least in part on the fourth commands between the driver and an SPTI; and (e) transmitting sixth commands based at least in part on the fifth commands between the SPTI and a drive.

[0017] Another embodiment of the invention comprises a method of sending write commands to a drive from a computer application which does not have administrator privileges, the method comprising the steps of: (a) transmitting write commands between a computer application which does not have administrator privileges and a driver programming interface; (b) transmitting second commands based at least in part on the write commands between the driver programming interface and a service proxy; (c) transmitting third commands based at least in part on the second commands between the service proxy and a service with administrator privileges; (d) transmitting fourth commands based at least in part on the third commands between the service with administrator privileges and a driver configured in SPTI mode; and (e) transmitting fifth commands based at least in part on the fourth commands between the driver and a drive.

[0018] Another embodiment of the invention comprises a program product for sending write commands to a drive from a computer application which does not have administrator privileges. The program product include machine-readable program code for causing, when executed, a machine to perform the following method: (a) transmitting write commands between a computer application which does not have administrator privileges and a driver programming interface; (b) transmitting second commands based at least in part on the write commands between the driver programming interface and a service proxy; (c) transmitting third commands based at least in part on the second commands between the service proxy and a service with administrator privileges; (d) transmitting fourth commands based at least in part on the third commands between the service with administrator privileges and a driver; (e) transmitting fifth commands based at least in part on the fourth commands between the

driver and an SPTI; and (f) transmitting sixth commands based at least in part on the fifth commands between the SPTI and a drive.

[0019] Reference will now be made in detail to various embodiments of the invention, which are illustrated in the drawings. An effort has been made to use the same reference numbers throughout the drawings to refer to the same or like parts.

[0020] Figure 4 shows a schematic diagram of a system 200 which enables an application without administrator privileges 102B to write to a drive without using a second, customized driver 112 working in conjunction with a IDE bus 110 and without using a third party driver 114. The embodiment shown describes a CD drive interface comprising a computer application without administrator privileges 102B, a driver program interface 116, a service proxy 118, a service with administrator privileges 120, and OS driver 104, an SPTI 106, and a CD drive 108.

[0021] The computer application without administrator privileges 102B may be, for example, a Windows application. This application 102B transmits commands (e.g., write commands) to the driver programming interface 116 by means of a high level language.

[0022] The driver programming interface 116 is the top level interface which uses the service proxy 118 to communicate with the CD drive 108. The driver interface 116 provides write/print specific commands that allow the imaging components to read information about the drive 108 and media, and to send settings, as well as actual image data to be printed, to the drive 108. The driver programming interface 116 interprets the high level language commands from the application 102B and communicates with the service proxy 118 by means of high level language (“HLL”) and/or application programming interface (“API”) language.

[0023] The service proxy 118 abstracts the need to use inter-process communication (“IPC”) mechanisms for an application without administrator privileges 102B to communicate with the service with administrator privileges

120. Upon recognizing the lack of administrator privileges, the proxy 118 provides generic commands to the service with administrator privileges 120.

[0024] The commands transmitted by the service proxy 118 direct the service with administrator privileges 120 to send an input, output, or status (e.g., is the drive running) commands to the drive 108. An input command is a command where the driver 104 wants to read some information from the drive 108 such as the media parameters. An output command is a command such as sending a track of pixel data to the drive 108. The service proxy 118 may also provide standard mechanisms to find all print capable drives on the local system, get a description and drive letter for the drives, send commands to open and close a particular drive for printing, send commands giving the driver 104 an exclusive lock on the drive 108, etc.

[0025] To send these commands, the service proxy 118 interfaces with the service with administrator privileges 120. The service with administrator privileges 120 provides administrator access to the driver 104 and is programmed to use the SPTI mode of the class driver 104. Accordingly, in communicating with the driver 104, the service with administrator privileges 120 uses SCSI-pass-through ("SPT") direct and/or IOCTL commands, i.e., the service with administrator privileges 120 is programmed to send corresponding low-level commands to the drive 108 via the SPTI 106.

[0026] As shown, the driver 104 and the SPTI 106 may be separate. In this case, the driver 104 communicates with the SPTI and the SPTI communicates with the drive 108 by means of SCSI commands. By way of contrast, the drive 104 could be combined in a driver 122 configured in SPTI mode in which case the driver 122 could communicate directly with the drive 108 by means of SCSI commands.

[0027] Similarly, as shown the service proxy 118 and the service with administrator privileges 120 may be separate. However, it should be recognized that the service proxy 118 and the service with administrator privileges 120 may be combined into one integrated device 124.

[0028] It also should be recognized that if the computer application is an application with administrator privileges 102A, the system 200 will work much like the system 100 shown in Figure 1A. In addition, implicit in this system 200 is an assumption that the firmware for the drive 108 is programmed to recognize the SCSI commands it receives as writing/printing commands.

[0029] The aforementioned system and method of transferring data to a disc provide numerous advantages. For example, the system allows an imaging system to communicate with an optical disc drive, without having to deal with the low-level complexities of SCSI drivers. By way of another example, the system does not require the user to have administrator privileges to access the drive through the SPTI 106. In addition, by using standard OS components and thereby eliminating the need for a new driver, the system: (a) does away with the licensing costs associated with using a third party CD-ROM driver that is not part of the operating system; and (b) does not require certification by the Microsoft Hardware Quality lab.

[0030] Although the aforementioned describes embodiments of the invention, the invention is not so restricted. It will be apparent to those skilled in the art that various modifications and variations can be made to the disclosed embodiments of the present invention without departing from the scope or spirit of the invention. Accordingly, these other systems and methods of transmitting write commands from an application without administrator privileges to a drive are fully within the scope of the claimed invention. Therefore, it should be understood that the apparatus and method described herein are illustrative only and are not limiting upon the scope of the invention, which is indicated by the following claims.

What is claimed is:

1. A system having a processor, comprising:

a drive operable, responsive to non-standard print commands, to print pixels of image data on a disc removably inserted in the drive;

a driver configured to communicate the non-standard print commands with the drive only if the non-standard print commands are received at the driver from a sender having administrator privileges;

a computer application that lacks administrator privileges configured to send high level language commands;

a driver programming interface configured to transmit the non-standard print commands via high level language and/or application programming interface function calls in response to the high level language commands received from the computer application;

a service proxy that receives the non-standard print commands from the driver programming interface;

a service having administrator privileges that receives the non-standard print commands from the service proxy and sends the print commands to the driver, wherein the driver is either in communication with a small-computer-system-interface-pass-through-interface (“SPTI”) or is configured in SPTI mode, and wherein the SPTI or the SPTI configured driver sends the non-standard print commands to the drive to print the disc based on the administrator privileges of the service.

2. The system according to claim 1, wherein the service proxy is configured to communicate with the service having administrator privileges by means of an inter-process communication (“IPC”).

3. The system according to claim 2, wherein the service having administrator privileges is configured to communicate with the driver using small-computer-system-interface-pass-through (“SPT”) direct and/or input/ output control (“IOCTL”) language.

4. The system according to claim 3, wherein if the driver is in communication with the SPTI, the driver is configured to communicate with the SPTI using small-computer-system-interface (“SCSI”) commands, and wherein if the driver is configured in SPTI mode, the driver is configured to communicate with the drive using SCSI commands.

5. The system according to claim 1, wherein the drive is an optical drive.

6. The system according to claim 5, wherein if the driver is in communication with the SPTI, the driver is configured to communicate with the optical drive using small-computer-system-interface (“SCSI”) commands via the SPTI.

7. The system of claim 1, wherein the system is a single personal computer, and wherein the computer application, the driver programming interface, the service proxy, the service, the driver, and the drive are disposed in the personal computer.

8. The system of claim 7, wherein the personal computer runs a Windows operating system.

9. The system of claim 1, wherein the service proxy determines if the computer application lacks administrator privileges.

10. The system of claim 9, wherein the service proxy provides the non-standard print commands to the service having administrator privileges if the computer application lacks administrator privileges.

11. The system of claim 1, wherein the service proxy is external to the drive.

12. The system of claim 1, wherein the driver program interface generates the non-standard print commands.

13. The system of claim 1, wherein the computer application that lacks administrator privileges generates the non-standard print commands.

14. A program product comprising a computer-readable medium that stores computer-readable program code for causing, when executed, a computer to perform the following method:

transmitting, between a computer application which does not have administrator privileges and a driver programming interface, high level language write commands for printing pixels of image data on a disc removably inserted in a drive;

transmitting non-standard second commands based at least in part on the write commands between the driver programming interface and a service proxy using high level language and/or application programming interface function calls;

transmitting non-standard third commands based at least in part on the second commands between the service proxy and a service having administrator privileges;

transmitting non-standard fourth commands based at least in part on the third commands between the service having administrator privileges and a driver, wherein the driver is configured to communicate with a small-computer-system-interface-pass-through-interface (“SPTI”) only if the non-standard fourth commands are received from a sender having administrator privileges;

based on the administrator privileges of the service, transmitting non-standard fifth commands based at least in part on the fourth commands between the driver and the SPTI; and

transmitting non-standard sixth commands based at least in part on the fifth commands between the SPTI and the drive to print the disc.

15. The program product according to claim 14, wherein the drive is an optical drive.