GEOTEXT Translations

translations@geotext.com
www.geotext.com

STATE OF CALIFORNIA)
)
)
COUNTY OF SAN FRANCISCO) ss

CERTIFICATION

This is to certify that the attached translation is, to the best of my knowledge and belief, a true and accurate translation from Japanese into English of the attached Patent Application No. 05-66937, with publication date of March 19, 1993.

Zach Duncan, Senior Project Manager
Geotext Translations, Inc.

A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

State of California, County of San Francisco
Subscribed and sworn to (or affirmed) before me
on this 30th day of April, 2018,
by Zach Duncan,
proved to me on the basis of satisfactory evidence

(19) JAPANESE PATENT OFFICE (JP)

(12) KOKAI TOKKYO PATENT JOURNAL (A)

(11) PATENT APPLICATION PUBLICATION NO. 05-66937

(43) Publication Date March 19, 1993

(51) Int. Cl.[X]:	Identification Codes:	Sequence Nos. for Office Use:	FI	Technical Disclosure Section
G06F 9/06 13/00	440 Q 351 L	8944-5B 7368-5B		

Examination Request: Not filed

No. of Claims: 3 (Total pages in original Japanese-language document: 19)

(21) Filing No.: H04-56215

(22) Filing Date: February 6, 1992

(31) Prior Number: H03-40936

(32) Priority Date: February 12, 1991

(33) Priority Country: Japan (JP)

(71) Applicant(s): 000000295
Oki Electric Industry, Co., Ltd.
1-7-12 Toranomon, Minato-ku, Tokyo

(72) Inventor(s): Tadao SHIMIZU,
Oki Electric Industry, Co., Ltd.
1-7-12 Toranomon, Minato-ku, Tokyo

(72) Inventor(s): Nobuhiro KUBOTA,
Oki Electric Industry, Co., Ltd.
1-7-12 Toranomon, Minato-ku, Tokyo

(72) Inventor(s): Hiroaki MIYAZAWA,
Oki Electric Industry, Co., Ltd.
1-7-12 Toranomon, Minato-ku, Tokyo

(74) Agent(s): Yuko SATO, Attorney

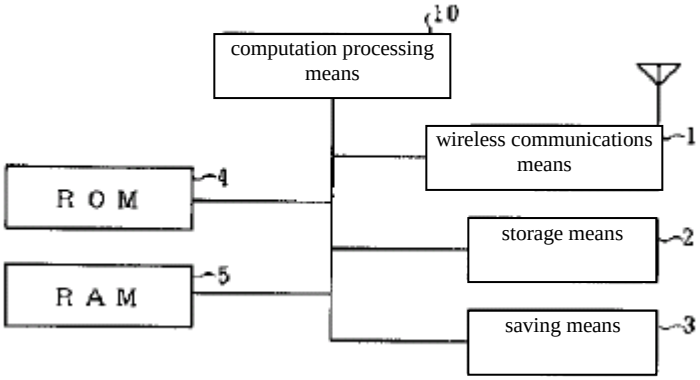
continued on last page

(54) [Title] Data Processing Device and Process Modification Method Therefor

(57) Abstract

[Objective] To reduce labor associated with modifying a program.

[Constitution] Data to be modified for the processing sequence of a data process is received by a wireless communications means 1. Next, the data that was received is saved in a saving means 3, and after its validity has been assessed, it is stored in a storage means 2. Consequently, in cases where there are a plurality of data processing devices having a program that must be modified, the operation can be easily performed even in cases where [said devices] are located in a dangerous area. In addition, it is possible to perform the operation even without needing to disassemble or assemble the data processing devices.



Embodiment of the data processing device of the present invention

1

[Claims]

[Claim 1] A data processing device comprising:

a storage means that stores a processing sequence for a data process;

a computation processing means that sequentially reads out said processing sequence from said storage means and executes [said sequence];

a wireless communication means that receives data for the part of said processing sequence that is to be modified; and

a saving means that saves said modification portion;

wherein,

the data that is received by said wireless communications means is saved in the saving means, and after its validity has been determined, [said data] is stored in said storage means.

[Claim 2] A data processing device that is furnished with an interface section that is connected to an external device, the data processing device being characterized in that it comprises of:

a nonvolatile read/write memory that can only be overwritten at certain times and that stores the processing sequence for the data process; and

a transfer means that transfers the data that was input from said interface section to said nonvolatile read/write memory.

[Claim 3] A data process modification method characterized in that:

data for the modification portion in the processing sequence for a data process is transmitted to one or more devices by means of either individual transmission or broadcast transmission;

the data for said modification portion is saved by each device on the receiving end, and error checking is performed; and

when there are no errors in the data of said modification portion,

the processing sequence for said data process is replaced with the data in said modification portion.

[Detailed Description of the Invention]

[0001]

[Field-of-use from the standpoint of manufacturing] The present invention pertains to a data processing device that stores a program in a storage means such as ROM and performs operations, as well as to a data process modification method that modifies the contents of that program.

[0002]

[Prior Art] Until now, in data processing devices that store a program in a ROM or other storage means and then perform operations, in cases where the program needs to be revised, in some cases, the printed circuit board on which the ROM was mounted has had to be replaced. Figure 2 is a block diagram of an example of a conventional data processing device. The device shown in the figure is a control device for a printer, which is a kind of data processing device. This device is made up of a CPU 51, ROM 52, RAM 53, and I/O port 54 and the like. The CPU 51 is the center that controls the printer. The ROM 52 stores text pattern data and programs to be executed when printing. The RAM 53 is used as a work area for programs and a reception buffer. The I/O port 54 is connected to drive mechanisms such as the print head 56, space motor (SP), and linefeed motor (LF) 57 and the like; it is also connected to switches and lamps on the operation panel 58 and an interface section 59 [for interfacing with] an external device 60. A bus line 55 is connected to each component, and sends and receives addresses and data. The external device 60

2

is made up of a personal computer or workstation main unit and a disk drive mechanism; it is connected to the printer shown in the figure via the interface section 59. In addition, in a data processing device operating as an online system, programs—that is, load modules—were replaced by the scheme shown in Figure 3 (see pages 177 - 179 in Unexamined Japanese Patent Application No. 62-280933).

[0003] Figure 3 is a block diagram that shows an online system in accordance with a conventional load module modification scheme. The system that is shown in the figure is composed of a load module modification management table 22, a load module library 24, and a main memory 26 and the like. The load module modification management table 22 manages and saves a load module identifier "A" that is input from the load module library 24 and modification information for that load module. The load module library 24 may, for example, be composed of a magnetic disk device or the like; it stores load modules contained in a data processing sequence in the online system.

[0004] The main memory 26 is made up of a random access memory (RAM) or the like; it loads load modules in the load module library 24. Load modules that have been loaded into the main memory 26 are sequentially read out by a processor (not shown in the figure) and executed. In such systems, the contents of the load module that is in main memory 26 are modified using modification information in the load module modification management table 22. The load module can then be executed using the contents that were modified. In addition, even if the online system experiences a failure, by following said load module modification management table 22, the modification contents of the load module in main memory 26 are safeguarded without loss of the modification information.

[0005]

[Issues To Be Resolved by the Invention] The above notwithstanding, with the above described conventional technology, the following problems arise. Specifically, using a conventional data processing device, in cases where there are multiple data processing devices whose programs are to be modified, one must replace the ROM circuit board for each device. In addition, in cases where there are multiple ROM circuit boards, one must search out the ROM circuit board that stores the program to be modified on each device, pull the located ROM circuit board out from its socket, and replace it. Consequently, a great deal of work is required for the modification operation. In particular, in cases where it is necessary to replace the ROM circuit board in a data processing device that is operating in a dangerous location or that is operating in a location where a person cannot easily access it, that much more labor will be required.

[0006] In addition, in order to actually replace the ROM circuit board, a large number of steps may be required, and if the person performing the work is not used to performing such a task, failures may easily occur, and operation safety can become an issue. Now, printers, which are used in a wide variety of computer systems as shown in Figure 2, constitute an example of a kind of data processing device. The program that controls the actual printing

operation has traditionally been stored in a ROM 52. Consequently, if something goes wrong with the program or there is a need to modify or improve printer usage, one must replace the ROM 52. During such replacement operations, one must first open the printer cover and possibly even disassemble the machinery. Either way, one requires an experienced operator or maintenance person for such replacement operations, and considerable time is required for the work. In addition, one must also prepare a new ROM for the replacement. Further still, in order to reduce the cost of the printer, a mask ROM may be used for the ROM 52, and in cases where a mask ROM is installed onto the circuit board while eliminating the IC socket, then replacing the ROM 52 becomes exceedingly difficult.

[0007] On the other hand, in a load module modification scheme for an online system like that shown in Figure 3, the following sorts of issues may arise. Specifically, in cases where the load module is stored not in RAM but in ROM, this sort of approach cannot be used. In addition, in cases where, outside of failure times, one starts the system back up after having cut off power to the system, then using the modification information that is stored in the load module modification management table 22, one must insert a patch once again into the load module that is in main memory 26. Consequently, if the amount of patching to be done is high, the task of modifying the load module will be troublesome, time will be required until the system starts back up, and throughput will be lowered.

[0008] The present invention was conceived taking the above issues into consideration. As such, it is an object of the present invention to provide a data processing device and a process modification method that aims to improve the operational safety and reduce the amount of labor in program modification work by performing program modification via a device interface or through remote operations that are performed using wireless or broadcast communications.

[0009]

[Means for Resolving the Issues Described] The data processing device of the present invention comprises: a storage means that stores a processing sequence for a data process; a computation processing means that sequentially reads out said processing sequence from said storage means and executes [said sequence]; a wireless communication means that receives data for the part of said processing sequence that is to be modified; and a saving means that saves said modification portion;

wherein, the data that is received by said wireless communications means is saved in the saving means, and after its validity has been assessed, [said data] is stored in said storage means. Another data processing device of the present invention constitutes a data processing device that is furnished with an interface section that is connected to an external device, the data processing device being characterized in that it comprises: a nonvolatile read/write memory that can only be overwritten at certain times and that stores the processing sequence for the data process; and a transfer means that transfers the data that was input from said interface section to said nonvolatile read/write memory.

[0010] The data process modification method of the present invention is characterized in that: data for the modification portion in the processing sequence for a data process is transmitted to one or more devices by means of either individual transmission or broadcast transmission; the data for said modification portion is saved by each device on the receiving end, and error checking is

performed; and when there are no errors in the data of said modification portion, the processing sequence for said data process is replaced with the data in said modification portion.

[0011]

[Effects] In the data processing device and process modification method therefor of the present invention, data for the modification target in a processing sequence of a data process is received by a wireless communications means or the like. Next, the data that was received is saved by a saving means, and after its validity has been determined, it is stored in a storage means. Consequently, in cases where there are multiple data processing devices for which a program must be modified, it is easy to perform the operation even when those devices are in a dangerous location. In addition, it is possible to perform the operation without disassembling or assembling the data processing device. Further still, in data processing devices that are furnished with an interface section that is connected to an external device, the processing sequence of the data process is stored in a nonvolatile read/write memory. Next, the data that has been input from said interface section at a certain time is transferred to the nonvolatile read/write memory by means of a transfer. As a result, overwriting of control programs for printers and the like is easily performed.

[0012]

[Embodiments] Below, embodiments of the present invention shall be described in detail with reference to the figures. Figure 1 is a block diagram of an embodiment of the data processing device of the present invention. The device in the figure is made up of a wireless communications means 1, storage means 2, saving means 3, ROM 4, and RAM 5 — these being connected to the bus line of a computation processing means 10. The wireless communications means 1 receives data that has been transmitted using electromagnetic waves. Using this wireless communications means 1, beyond individual communications involving a single recipient, it is also possible to handle broadcast communications involving multiple recipients. The storage means 2 is made from EEPROM or the like. As is well known, EEPROM is a kind of memory that operates as a ROM that cannot normally be overwritten, though it can be overridden using electronic means. A program that is actually operated by the computation processing means 10 is stored in this storage means 2. Patches are attached to the program that is stored in this storage means 2.

[0013] The saving means 3 temporarily saves patch programs that have been transmitted via the wireless communications means 1. This saving means 3 is made up of a battery backup memory (BBM) or the like. The computation processing means 10 performs error control on patch programs, and in cases where there are no errors, it expands the patch into the storage means 2. The ROM 4 is a so-called read-only memory; it stores all of the programs that are used by the terminal. The program that is stored in this ROM 4 is the original program prior to modification. When power is turned onto the device, after an initial diagnostic process is executed, the program is transferred from the ROM 4 to the storage means 2. The RAM 5 is a so-called random access memory that can be overwritten. This RAM 5 is used as a work area for the computation processing means 10 that executes the program.

[0014] The original program that is stored in ROM 4 is divided into a portion that is executed in ROM 4 and a portion that is

executed on the storage means 2. The former is referred to as a core, while the latter is referred to as firmware. The core performs such activities as executing the initial diagnosis performed when the power is switched on, transferring firmware to the storage means 2, and expanding patches in the storage means 2. The firmware is a program executed on the terminal. This firmware can be overwritten on the storage means 2. The firmware that is stored in ROM 4 is a program that has a version number. This firmware is only transferred to the storage means 2 once when the power is turned on. When the firmware version is updated, then a patch is applied to the storage means 2. At this time, the version number record section on the storage means 2 is updated. Subsequently, at the same time that the patch is applied to the storage means 2, each time the version is being updated, the version number is also updated. This flow of control is shown in Figure 3.

[0015] Figure 4 is a flowchart that shows a processing sequence for modifying a program. First, when the power is turned on a determination is made as to whether the installed ROM 4 is new (step S21). This determination may be made by, for example, pressing a certain key on the keyboard while the power is turned on. If the ROM 4 is new, then the firmware is transferred to the storage means 2 (step S22). If it is not, then because the firmware that is stored in the storage means 2 is the latest version, the transfer of firmware from ROM 4 is not performed.

[0016] Patch data that has been received from the wireless communications means 1 is temporarily saved in the saving means (step S23) without immediately being expanded into the storage means 2. When reception of the latest patch data has been completed (step S24), control privileges are transferred to the ROM 4, and the patch is expanded into the storage means 2 (step S25). At this point, the reason that control privileges are temporarily transferred to ROM is so as to make the entire region of the storage means 2 eligible for patching. Now, thanks to the execution of step S36, in cases where the firmware of the storage means 2 is transferred to the RAM 5, when the patch is expanded into the storage means 2, there is no need to transfer the control privileges to the core of the ROM 4. In other words, the patch can be expanded into the storage means 2 with the control privileges remaining as is in the firmware in RAM 5.

[0017] Figure 5 is an explanatory diagram of a sequence for implementing the data process modification method of the present invention. First, the patch transmission device transmits patch data to a plurality of data processing devices that are within the reach of [its] electromagnetic waves through broadcast transmissions using a wireless communications means 1 (T1). Next, after an elapse of time (T2) that is needed for each data processing device to expand the patch into the storage means 2, a patch transmission device sends a patch result read command by means of broadcast communications (T3). Each data processing device transmits a total hash value that is set in the patch data and a response that determines whether the patch terminated normally based on the patch version number to the patch transmission device (T4). In cases where the patch terminated abnormally, then once again the patch operation is repeated only for that data processing device (T5).

[0018] Figure 6 is an explanatory diagram of the transition of

control privileges. Due to the power being turned on, the data processing device passes through each of the following device modes. These device modes are an initial diagnostic M1, transfer of the firmware in ROM to EEPROM M2, operation M3, patch data incoming M4, patch saving to EEPROM M5, and operation M6. In each of these device modes, control privileges are transitioned between the ROM and the EEPROM. First, in the initial diagnosis M1 and the transition of the firmware section in ROM to EEPROM M2, control privileges reside in the core on ROM. In the initial diagnostic M1, the ROM and RAM are checked, and a hash check is performed for the EEPROM. The transfer of the firmware section in ROM to EEPROM is executed only when there is an instruction to the effect that a new ROM has been installed.

[0019] Next, during operation M3 and patch data incoming M4, the control privileges are transitioned from ROM to EEPROM and reside in the firmware on EEPROM. In operation M3, operation task processing is performed. In this operation M3, if a patch is received, then processing advances to patch data incoming M4. As a result, processing advances to storing patch in EEPROM M5, and control privileges are transitioned from EEPROM to ROM. After that, once processing moves to operation M6, control privileges are once again transitioned from ROM to EEPROM.

[0020] Figure 7 is an explanatory diagram of a transfer when the power is turned on. The program that is stored in ROM is made up of a core 41 and a firmware 42. The core 41 is left on ROM without being forwarded. The firmware 42 is the portion that is forwarded to the EEPROM, and contains a version number record portion 43 for said program. Forwarding of this firmware 42 is only performed once. In other words, the version number that is forwarded from the ROM constitutes the initial value of the version number in EEPROM. When executing a patch, a comparison is made between the version number that is set in the last patch command shown in Figure 10 prior to execution and the version number that is stored in the EEPROM, and the patch is stored in the EEPROM only if these version numbers are the same.

[0021] Figure 8 is an explanatory diagram of a patch command scheme. The commands in the figure consist of an identifier CMD, command length LNG, flags FLG, patch address ADR, and patch data DATA. The identifier CMD is a code for identifying other commands and the patch command. Command length LNG shows the total number of bytes in said patch command. The flags FLG show which portion of the patch command prior to partitioning the segmented patch command is located in cases where the patch command is segmented into multiple patch commands and then transmitted. The patch address ADR is the address of the portion where the patch data was entered into the program. The patch data DATA is the patch data constituting the revised portion of the program.

[0022] Figure 19 is an explanatory diagram of the contents of the flags. The flags shown in the figure are composed of two-bit data. "00" indicates that the segmented patch command is one that comes from the middle of the patch. "01" indicates that the

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.