

{Description}

{Title of Invention} SYSTEM EVENT LOGGING SYSTEM

{Technical Field}

{0001}

The present invention relates to a system event logging system which records a log of a system event related to monitoring target process.

{Background Art}

{0002}

PTL 1 ("a computer system and an application program operation reproducing method") discloses "a computer system and an application program operation reproducing method which, when an application program abnormally ends, can correctly reproduce a state of the application program as in an operation upon an abnormal end of the application program without applying a redundant load for reproducing the state to the application program, and can substantially reduce an operation load of the state reproducing operation and an operating time."

{0003}

More specifically, an operation is recorded to reproduce the operation.

{0004}

When a log of a system event is recorded to record the operation, many unnecessary logs are included, and therefore it is difficult in some cases to reproduce the operation from the log.

{Citation List}

{Patent Literature}

{0005}

{PTL 1} JP 2002-024055 A

{Summary of Invention}

{Technical Problem}

{0006}

An object is to remove non-target system events, and select and record a system event required to reproduce an operation.

{Solution to Problem}

{0007}

A system event logging system according to the present invention has the following elements including: (1) a decision target event acquiring part which sequentially acquires a system event related to decision target

process; (2) a filter list which stores a filter record which associates an event condition, a flag condition, a log write instruction and a flag operation instruction for each filter; (3) a flag memory part which stores a flag value; and (4) a filtering part which repeats processing of sequentially reading a filter record of each of the acquired system event, deciding whether or not the system event satisfies the event condition for each read filter record, deciding whether or not the flag value satisfies the flag condition when the flag condition is further set, writing the system event as a log according to the log write instruction when the event condition and the flag condition are satisfied, and updating the flag value according to the flag operation instruction when the flag operation instruction is further set.

{0008}

The filter record is further associated with the screen image acquisition instruction, and the filtering part records a screen image according to the screen image acquisition instruction when the event condition and the flag condition are satisfied and the screen image acquisition instruction is set.

{0009}

A program according to the present invention causes a computer which serves as a system event logging system having: a filter list which stores a filter record which associates an event condition, a flag condition, a log write instruction and a flag operation instruction for each filter; and a flag memory part which stores a flag value to execute the following steps including: (1) a decision target event acquiring step of sequentially acquiring a system event related to decision target process; and (2) a filtering step of repeating processing of sequentially reading a filter record of each of the acquired system event, deciding whether or not the system event satisfies the event condition for each read filter record, deciding whether or not the flag value satisfies the flag condition when the flag condition is further set, writing the system event as a log according to the log write instruction when the event condition and the flag condition are satisfied; and updating the flag value according to the flag operation instruction when the flag operation instruction is further set.

{Advantageous Effect of Invention}

{0010}

When a filter is adapted by providing a flag condition and a flag

operation instruction for each filter and using a flag condition as a condition for adapting a filter, the flag is operated according to the flag operation instruction, so that it is possible to interface filters through the flag. Further, it is also possible to realize interface processing through the flag even during processing of the system event.

{0011}

A screen image acquisition instruction is provided to the filter, so that it is possible to adequately record a screen image in response to an occurrence of the system event.

{Brief Description of the Drawings}

{0012}

{Fig. 1} Fig. 1 illustrates a view that illustrates operating environment of a system event logging system;

{Fig. 2} Fig. 2 illustrates a view that illustrates a processing flow of a logger unit;

{Fig. 3} Fig. 3 illustrates a view that illustrates a configuration of generating an internal process list;

{Fig. 4} Fig. 4 illustrates a view that illustrates an internal process list generation processing flow;

{Fig. 5} Fig. 5 illustrates a view that illustrates a configuration of acquiring a decision target event;

{Fig. 6} Fig. 6 illustrates a view that illustrates a decision target event acquisition processing flow;

{Fig. 7} Fig. 7 illustrates a view that illustrates a configuration of filtering;

{Fig. 8} Fig. 8 illustrates a view that illustrates a filtering processing flow;

{Fig. 9} Fig. 9 illustrates a view that illustrates a configuration of a system event;

{Fig. 10} Fig. 10 illustrates a view that illustrates a configuration of a filter list;

{Fig. 11} Fig. 11 illustrates a view that illustrates a configuration of event conditions;

{Fig. 12} Fig. 12 illustrates a view that illustrates a configuration of flag conditions;

{Fig. 13} Fig. 13 illustrates a view that illustrates a configuration of a flag operation instruction;

{Fig. 14} Fig. 14 illustrates a view that illustrates a configuration of a

screen image acquisition instruction;

{Fig. 15} Fig. 15 illustrates a view that illustrates a configuration of a log write instruction;

{Fig. 16} Fig. 16 illustrates a view that illustrates a log record (1/2);

{Fig. 17} Fig. 17 illustrates a view that illustrates a log record (2/2);

{Fig. 18} Fig. 18 illustrates a view that illustrates a configuration of outputting a log file;

{Fig. 19} Fig. 19 illustrates a view that illustrates an end process monitoring processing flow;

{Fig. 20} Fig. 20 illustrates a view that illustrates a configuration of a viewer unit;

{Fig. 21} Fig. 21 illustrates a view that illustrates a configuration of a viewer filter list;

{Fig. 22} Fig. 22 illustrates a view that illustrates a processing flow of a viewer unit; and

{Fig. 23} Fig. 23 illustrates a view that illustrates a hardware configuration of the system event logging system;

{Description of Embodiments}

{0013}

Fig. 1 illustrates a view that illustrates operating environment of a system event logging system. A monitoring target program 104 and a non-monitoring target program 105 operate by acquiring a system event which occurs in response to a user's operation of a keyboard or a mouse according to an API call. The logger unit 101 acquires this system event by means of an event cue 107 in the operation system 106 using a global hook, and selects only predetermined events among events which occur upon an operation of the monitoring target program 104 and store the predetermined events in a log file memory unit 102. Further, the logger unit 101 operates to adequately acquire and store a screen image in a screen image file memory unit 103.

{0014}

The log file memory unit 102 and the screen image file memory unit 103 are provided in, for example, a memory area of a hard disk device. Further, the logger unit 101, the monitoring target program 104 and the non-monitoring target program 105 are configured to operate by being loaded to a memory and having a program code sequentially read and

executed by a computing device.

{0015}

Hereinafter, an operation of the logger unit (logger unit) will be described. Fig. 2 illustrates a view that illustrates a processing flow of the logger unit. When the logger unit 101 is activated, internal process list generation processing (S201) is performed as preprocessing. By this means, the internal process list is generated. Details will be described below using Figs. 3 and 4. Next, in decision target event acquisition processing (S202), system events of the monitoring target program 104 and the non-monitoring target program 105 which are operating are acquired, and a decision target event is selected. Details will be described below using Figs. 5 and 6. Subsequently, in filtering processing (S203), an operation is performed of extracting an event according to a filter list and accumulating the event in an internal buffer as a log. Details will be described below using Figs. 7 to 17. Further, until an end instruction is received (S204), the decision target event acquisition processing (S202) and the filtering processing (S203) are performed. When the end instruction is received (S204), processing of outputting the log accumulated in the internal buffer as a log file is performed in log output processing (S205). Details will be described below using Fig. 18. Further, in addition to the processing in Fig. 2, end process monitoring processing is operating as another asynchronous task. Details will be described below using Fig. 19.

{0016}

First, the internal process list generation processing (S201) will be described. Fig. 3 illustrates a view that illustrates a configuration of generating an internal process list. The logger unit 101 has an internal process list generating unit 301, a monitoring target process list 302 and an internal process list 303. The internal process list generating unit 301 performs processing of acquiring an operation process list from an OS, and registering operating process corresponding to the monitoring target process (corresponding to the monitoring target program 104) stored in the monitoring target process list 302, in the internal process list 303. Processes corresponding to one or a plurality of monitoring target programs are registered in the monitoring target process list 302 in advance.

{0017}

Fig. 4 illustrates a view that illustrates an internal process list

generation processing flow. First, an operation process list is acquired from an OS (S401), and the following processing is repeated for each operation process included in the operation process list (S402). When operation process corresponds to one of monitoring target processes in the monitoring target process list 302 (S403), the operation process is added to the internal process list (S404). Further, a process start log is added to the internal buffer (S405). Furthermore, processing ends when all operation processes are processed (S406). By this means, the process of the monitoring targets which has been already activated and which is operating is registered in the internal process list.

{0018}

In a process start log, a recording date and, in addition, "start application" as an event type, a specific name of a process name, a specific value of a process ID, "application" as an operation target value and "process" as a class name of the operation target are recorded. A configuration of a log will be described below using Fig. 16.

{0019}

Next, the decision target event acquisition processing (S202) will be described. Fig. 5 illustrates a view that illustrates a configuration of acquiring a decision target event. The logger unit 101 has a decision target event acquiring unit 501 and an internal buffer 502 in addition to the above internal process list 302 and the monitoring target process list 303. The decision target event acquiring unit 501 acquires a system event from an event cue in order of occurrence, and extracts a decision target event. Further, when the monitoring target process is newly activated, processing of adding this process to the internal process list 303 is performed.

{0020}

Fig. 6 illustrates a view that illustrates a decision target event acquisition processing flow. First, a system event is acquired from an event cue utilizing a global hook (S601). Further, process of the event is specified (S602). Whether or not the specified process is an event of the monitoring target process (S603) and, when the process does not correspond to one of the monitoring target processes included in the monitoring target process list 302, the oldest system event of the event cue is erased (S604), and the step returns to processing of S601 again. By contrast with this, when the process corresponds to one of the monitoring target processes, whether or not

this process is registered in the internal process list is decided (S605). When the process is not registered, this process is added to the internal process list (S606). Further, a process start log is added to the internal buffer (S607). On the other hand, when the process is already registered, processing is then finished.

{0021}

Subsequently, the filtering processing (S203) will be described. Fig. 7 illustrates a view that illustrates a configuration of filtering. The logger unit 101 has a filtering unit 701, a filter list 702 and a flag memory unit 703 (a global flag memory unit 704 and a local flag memory unit 705) in addition to the above screen image file memory unit 103 and the internal buffer 502. In the filter list 702, a plurality of filters are stored in order of processing. The filtering unit 701 sequentially reads filters, compares conditions of the decision target event and conditions of a filter according to content of the filter, decides the flag of the local flag memory unit 705, updates the flag in the local flag memory unit 705, stores a screen image in the screen image file memory unit 103, and additionally writes content of an event in the internal buffer 502 as a log.

{0022}

The global flag memory unit 704 of the flag memory unit 703 is a flag which is commonly operated by each process or is compared. By contrast with this, the local memory unit 705 is a flag to which one process is dedicated. Further, the global flag memory unit 704 is configured to have a plurality of flags, and be specified, operated or compared according to a global flag ID. Similarly, the local flag memory unit 705 is also configured to have a plurality of flags, and be specified, operated and compared according to a local flag ID.

{0023}

Fig. 8 illustrates a view that illustrates a filtering processing flow. Filter records are sequentially read from the filter list 702, and the following processing for a decision target event is performed (S801). First, in filter adaptation condition decision processing, whether or not a decision target event adapts to filter adaptation conditions is decided (S802).

{0024}

Hereinafter, configurations of a system event and a filter list will be described. Fig. 9 illustrates a view that illustrates a configuration of the

system event. The system event includes items of an event type: EventID, an operation parameter: Params, an operation target type: ElementTypeID, an operation target state: Status, an operation target name: ElementName, a class name of the operation target: ClassName, a unique ID of the operation target: ControlID, a caption of a root window of the operation target: RootName, a class name of the root window of the operation target: RootClassName and a process name: ProcessName.

{0025}

Fig. 10 illustrates a view that illustrates a configuration of the filter list. A filter record is provided for each filter in processing order, and the filter has items of a filter ID, a filter type, filter adaptation conditions (event conditions and flag conditions) and filter defining operations (a screen image acquisition instruction, a command instruction, a log write instruction and a flag operation instruction).

{0026}

In filter adaptation condition decision processing (S802), whether or not the decision target event adapts to the filter adaptation conditions is decided. It is decided that the decision target event is adaptive when the decision target event matches with the event conditions and matches with the flag conditions. In addition, when the flag conditions are not set, decision is made only for the event conditions.

{0027}

Decision of the event conditions will be described. Fig. 11 illustrates a view that illustrates a configuration of the event conditions. Conditions are configured to be set for each item of the system event. Except the item of no condition, each item is decided according to AND conditions. It is also possible to specify a left-hand match, a right-hand match and a partial match in addition to a perfect match.

{0028}

Decision of the flag conditions will be described. Fig. 12 illustrates a view that illustrates a configuration of the flag conditions. Except an item of no setting, when the global flag conditions and the local flag conditions are adaptive, it is decided that the overall flag conditions are adaptive. The global flag conditions include a global flag ID, comparison conditions and a comparison value. A value of a flag specified according to the global flag ID is read from the global flag memory unit 704 and, when the flag value

satisfies the comparison conditions compared to the comparison value, it is decided that the global conditions are adaptive. Conditions such as "equal", "not equal", "equal to or more than", "equal to or less than", "smaller than" or "higher than" can be set to the comparison conditions. Similarly, the local flag conditions include a global flag ID, comparison conditions and a comparison value, and, when a value of a flag specified according to a local flag ID is read from the local flag memory unit 705 and the flag value satisfies the comparison conditions compared to the comparison value, it is decided that the local flag conditions are adaptive. The same applies to the comparison conditions.

{0029}

When the conditions are decided as maladaptive based on these decisions according to filter adaptation condition decision processing (S802), the step returns to S801 to proceed to processing of the next filter record. When the conditions are decided as adaptive (S802), the step proceeds to processing for each filter type. When the filter type is "ignore" (S803), the step returns to S801 to proceed to processing of the next filter record.

{0030}

When the filter type is "non-operation" (S804), filtering processing ends without returning to S801. When the filter type is "only flag operation" (S805), flag operation processing (S806) is performed, the oldest event of the event cue is erased (S812) and filtering processing ends. When the filter type is "continuing operation" (S807), flag operation processing (S808) and filter defining operation execution processing (S809) are performed, and the step returns to S801 to proceed processing of the next filter record. When the filter type is "final operation" (S807), flag operation processing (S810) and filter defining operation execution processing (S811) are performed, the oldest event of the event cue is erased (S812), and filtering processing is finished.

{0031}

The above flag operation processing will be described. According to this processing, a flag operation instruction of a filter is executed. Fig. 13 illustrates a view that illustrates a configuration of a flag operation instruction. The flag operation instruction is configured with the global flag operation and the local flag operation, and is directed to performing the instructed flag operation except no setting. The global flag operation

includes a global flag ID, operation computation and an operation value. A flag value specified according to the global flag ID is read from the global flag memory unit 704, the operation value is computed for the read flag value, and a computation result is written in the flag value specified according to the global flag ID. In the operation computation, computation such as substitution, addition, subtraction, multiplication or division can be set. Similarly, the local flag operation also includes a local flag ID, operation computation and an operation value, and is directed to reading a flag value specified according to the local flag ID from the local flag memory unit 705, computes the operation value for the read flag value and writes the computation result in the flag value specified according to the local flag ID. The flag value is updated in this way.

{0032}

The above filter defining operation execution processing will be described. In this processing, processing of a screen image acquisition instruction, processing of a command instruction and processing of a log write command included in a filter defining operation of the filter are executed. No processing is performed according to each instruction in case of no setting.

{0033}

Fig. 14 illustrates a view that illustrates a configuration of the screen image acquisition instruction. The screen image acquisition instruction includes items of a capture scheme: SnapshotType, a capture image file format: SnapshotFormat, a range coordinate of partial capture: TargetRect, a compression rate of a Jpg format: JpegQuality, a capture timing delay time (ms): Delay, visibility/invisibility of window display check: IsCheckVisible, and acquisition of an image from a GUI cache of a system: IsUseGUICache. According to conditions of these items, a screen image is acquired through an OS, and is stored in the screen image file memory unit 103 as a screen image file.

{0034}

Identification information of a hot key can be set to an item of a command instruction of a filter. In processing of a command instruction, identification information of a hot key is read from the item of this command instruction, and an operation matching this hot key is activated through the OS.

{0035}

Fig. 15 illustrates a view that illustrates a configuration of a log write instruction. In the write type, one of "not-corrected", "corrected" and "not-corrected and corrected" can be set. In case of "not-corrected", information of each item of the system event is written in the internal buffer 502 as a log. In case of "corrected", an EventID correction value, an EventTypeID correction value, an EventName correction value and a Value correction value are written in each corresponding item as logs except an item of no setting, and an item value of a system event is written in other items. Further, in case of "not-corrected and corrected", two logs of a log corresponding to "not-corrected" and a log corresponding to "corrected" are recorded.

{0036}

Meanwhile, a configuration of a log record will be described. Figs. 16 and 17 illustrate views that illustrate a configuration of a log record. The log record includes items of a log ID: LogID, a recording date: DateTime, a user ID: UserID, an event type: EventID, an operation parameter: Params, a process name: ProcessName, a process ID: ProcessID, an operation target value: Value, an operation target type: ElementTypeID, an operation target state: Status, an operation target name: ElementName, a class name of the operation target: ClassName, an ID of a child item of the operation target: ChildID, right and wrong of a top window of the operation target: IsTopWindow, a control ID of the operation target: Rect, a rectangular range of the operation target (screen coordinate): RootRect, handle of the operation target: Handle, handle of the operation target: RootHandle, handle of a root window of the operation target: LinkImage, a file name of a relevant screen: LinkImage, a type of a capture screen (range): SnapshotType, a unique ID of the operation target: ControlID, a class name of a root window of the operation target: RootClassName, a caption of the root window of the operation target: RootName, an end: End, and a comment: Comment. Items which are not included in the system event can be effectively acquired as necessary through, for example, the OS and stored.

{0037}

The log output processing (S205) of outputting logs accumulated in the above processing will be described. Fig. 18 illustrates a view that illustrates a configuration of outputting a log file. The logger unit 101 has a

log file output unit 1801 in addition to the above log file memory unit 102 and the internal buffer 502. The log file output unit 1801 reads a log list including a series of logs from the internal buffer 502, and stores the log list in a file format in the log file memory unit 102.

{0038}

The logger unit 101 further performs end process monitoring processing according to an asynchronous task. According to this processing, that process has ended is recorded in the log. Fig. 19 illustrates a view that illustrates an end process monitoring processing flow. When the end process is acquired from the OS (S1901), the end process is erased from the internal process list 303 (S1902). Further, the process end log is added to the internal buffer (S1903).

{0039}

In the process end log, a recording date and, in addition, "application end" as an event type, a specific name of a process name, a specific value of a process ID, "application" as an operation target value and "process" as a class name of the operation target are recorded. This log is also recorded in the log file memory unit 102 as part of the log list.

{0040}

The log file and the screen image file accumulated in the above processing can be displayed by the viewer unit. Fig. 20 illustrates a view that illustrates a configuration of the viewer unit. A viewer unit 2001 reads the log file from the log file memory unit 102, reads a screen image file from the screen image file memory unit 103, and outputs, for example, displays the screen image file according to an instruction of a viewer filter list 2002.

{0041}

Fig. 21 illustrates a view that illustrates a configuration of a viewer filter list. A record is provided for each filter, and a filter ID, filter adaptation conditions and a display control instruction are associated and stored. The filter adaptation conditions are set for each item of a log. In the display control instruction, an instruction for displaying the log is set.

{0042}

Fig. 22 illustrates a view that illustrates a processing flow of the viewer unit. The following processing is repeated for each log record of a log file of the log file memory unit 102 (S2201). When the following processing is repeated for each filter record (S2202) and the log adapts to the filter

adaptation conditions (S2203), display processing related to the log is executed according to the display control instruction (S2204). In this case, when a screen display instruction is included, a screen image file is read and displayed according to this instruction. Further, if all filters are processed (S2205) and an end instruction is not received (S2206), the step returns to S2201 to proceed to processing related to the next log. These processings end at a point of time when these processings are applied to all logs (S2207). When the end instruction is received, processing ends at this point of time.

{0043}

In addition, the operation of the viewer unit can be executed by a computer separate from the logger unit. In this case, a log file and a screen image file recorded in the logger unit are duplicated to a computer of a viewer unit, and referred to.

{0044}

The system event logging system is a computer, and each element can execute processing according to a program. Further, it is possible to store the program in a storage medium and make a computer read the program from the storage medium.

{0045}

A hardware configuration of the system event logging system will be described. Fig. 23 illustrates a view that illustrates the hardware configuration of the system event logging system. A computing device 2301, a data memory device 2302, a memory 2303, a communication interface 2304, a data input device 2305 and a data output device 2306 are connected to a bus. The data memory device 2303 is, for example, a ROM (Read Only Memory) or a hard disk. The memory 2303 is generally a RAM (Random Access Memory). The program is generally stored in the data memory device 2302, and is sequentially read by the computing device 2301 in a state where the program is loaded to the memory 2303 to perform processing. The communication interface 2304 is used for communication through a network. The data input device 2305 is used to input data. The data output device 2306 is used to output (display or print) data.

{Reference Signs List}

{0046}

101 logger unit

102 log file memory unit

103	screen image file memory unit
104	monitoring target program
105	non-monitoring target program
106	operation system
107	event cue
301	internal process list generating unit
302	monitoring target process list
303	internal process list
501	decision target event acquiring unit
502	internal buffer
1801	log filer output unit
2001	viewer unit
2002	viewer filter list
2301	computing device
2302	data memory device
2303	memory
2304	communication interface
2305	data input device
2306	data output device

{Claims}

{Claim 1}

A system event logging system comprising:

(1) a decision target event acquiring part which sequentially acquires a system event related to decision target process;

(2) a filter list which stores a filter record which associates an event condition, a flag condition, a log write instruction and a flag operation instruction for each filter;

(3) a flag memory part which stores a flag value; and

(4) a filtering part which repeats processing of sequentially reading a filter record of each of the acquired system event, deciding whether or not the system event satisfies the event condition for each read filter record, deciding whether or not the flag value satisfies the flag condition when the flag condition is further set, writing the system event as a log according to the log write instruction when the event condition and the flag condition are satisfied, and updating the flag value according to the flag operation instruction when the flag operation instruction is further set.

{Claim 2}

The system event logging system according to claim 1, wherein:

the filter record is further associated with the screen image acquisition instruction; and

the filtering part records a screen image according to the screen image acquisition instruction when the event condition and the flag condition are satisfied and the screen image acquisition instruction is set.

{Claim 3}

A program causing a computer which serves as a system event logging system comprising: a filter list which stores a filter record which associates an event condition, a flag condition, a log write instruction and a flag operation instruction for each filter; and a flag memory part which stores a flag value to execute:

(1) a decision target event acquiring step of sequentially acquiring a system event related to decision target process; and

(2) a filtering step of repeating processing of sequentially reading a filter record of each of the acquired system event, deciding whether or not the

system event satisfies the event condition for each read filter record, deciding whether or not the flag value satisfies the flag condition when the flag condition is further set, writing the system event as a log according to the log write instruction when the event condition and the flag condition are satisfied; and updating the flag value according to the flag operation instruction when the flag operation instruction is further set.