(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2005-293016
(P2005-293016A)
(43)公開日　平成17年10月20日(2005.10.20)

(51)Int.Cl.[7]	F I		テーマコード（参考）
G06F 11/30	G06F 11/30	J	5B042
G06F 11/34	G06F 11/34	L	

審査請求 有　請求項の数 7　OL　（全 11 頁）

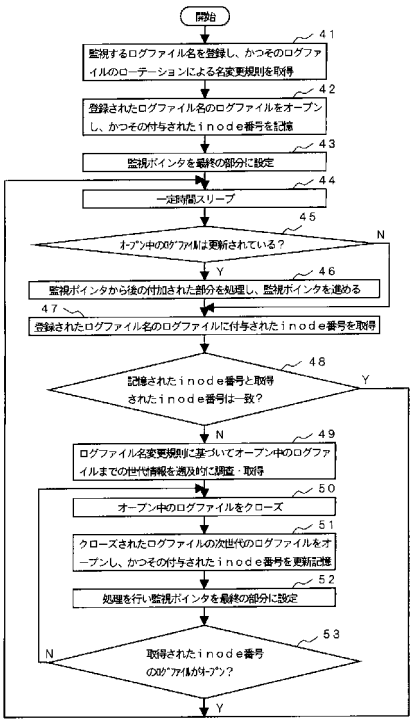
(21)出願番号　特願2004-104664 (P2004-104664)
(22)出願日　平成16年3月31日 (2004.3.31)

(71)出願人　301063496
東芝ソリューション株式会社
東京都港区芝浦一丁目１番１号
(71)出願人　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　100077849
弁理士　須山　佐一
(72)発明者　澤崎　武
東京都港区芝浦一丁目１番１号　東芝ソリューション株式会社内

Fターム(参考)　5B042 MA15

(54)【発明の名称】ログファイル監視プログラム、ログファイル監視方法、計算機システム

(57)【要約】
【課題】ログファイルを監視するためのログファイル監視プログラム、方法、および計算機システムにおいて、ログファイルのローテーションに適切に対応すること。
【解決手段】監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶し、第１の不変情報が記憶された状態において、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得し、記憶された第１の不変情報と取得された第２の不変情報とが同一であるか否かを判定し、判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更規則に基づいて監視中でオープン中のログファイルに至る世代情報を調査・取得し、取得された世代情報に基づいて監視中でオープン中のログファイルより新しい世代のログファイルを順にオープン対象として処理する。
【選択図】図４



【特許請求の範囲】
【請求項１】
監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶するステップと、
監視中でオープン中の前記ログファイルを処理するステップと、
前記第１の不変情報が記憶された状態において、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得するステップと、
前記記憶された第１の不変情報と前記取得された第２の不変情報とが同一であるか否かを判定するステップと、
前記判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更規則に基づいて監視中でオープン中の前記ログファイルに至る世代情報を調査・取得するステップと、
前記取得された世代情報に基づいて監視中でオープン中の前記ログファイルより新しい世代のログファイルを順にオープン対象として処理するステップと
をコンピュータに実行させるためのログファイル監視プログラム。
【請求項２】
前記不変情報が、ｉｎｏｄｅ番号であることを特徴とする請求項１記載のログファイル監視プログラム。
【請求項３】
前記処理が、障害発生の検知であることを特徴とする請求項１記載のログファイル監視プログラム。
【請求項４】
前記ログファイル名変更規則が、前記監視すべきログファイル名に「．ｎ」（ｎは整数で世代が古くなると１ずつ増加）を付加する規則であると仮定することを特徴とする請求項１記載のログファイル監視プログラム。
【請求項５】
前記世代情報が、ログファイル名と該ログファイル名のログファイルに付与された不変情報とからなる組の情報を前記第２の不変情報から前記第１の不変情報までたどったものであることを特徴とする請求項１記載のログファイル監視プログラム。
【請求項６】
監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶するステップと、
監視中でオープン中の前記ログファイルを処理するステップと、
前記第１の不変情報が記憶された状態において、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得するステップと、
前記記憶された第１の不変情報と前記取得された第２の不変情報とが同一であるか否かを判定するステップと、
前記判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更規則に基づいて監視中でオープン中の前記ログファイルに至る世代情報を調査・取得するステップと、
前記取得された世代情報に基づいて監視中でオープン中の前記ログファイルより新しい世代のログファイルを順にオープン対象として処理するステップと
を具備することを特徴とするログファイル監視方法。
【請求項７】
監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶する手段と、監視中でオープン中の前記ログファイルを処理する手段と、前記第１の不変情報が記憶された状態において、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得する手段と、前記記憶された第１の不変情報と前記取得された第２の不変情報とが同一であるか否かを判定する手段と、前記判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更

規則に基づいて監視中でオープン中の前記ログファイルに至る世代情報を調査・取得する手段と、前記取得された世代情報に基づいて監視中でオープン中の前記ログファイルより新しい世代のログファイルを順にオープン対象として処理する手段とを備えたログファイル監視部と、

前記ログファイル監視部により監視される前記各ログファイルを記憶するログファイル記憶部と

を具備することを特徴とする計算機システム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、ログファイルを監視するためのログファイル監視プログラム、ログファイル監視方法、およびそのようなプログラムまたは方法が適用され得る計算機システムに係り、特に、ログファイルのローテーションに適切に対応するログファイル監視プログラム、ログファイル監視方法、および計算機システムに関する。

【背景技術】

【０００２】

ログファイルを監視（周期的に参照）し特定のイベントの発生を検知するような処理を行うプログラムでは、一般的に、ある監視周期における監視で、前回の監視周期で処理された後に付加された部分のみを処理対象とする。付加された部分は、例えば、前回の監視周期のとき最終処理部分を監視ポインタで指し示しておきその監視ポインタの後方として認識することができる。

【０００３】

一方、ログファイルの監視とは全く独立した機構として、ログファイルのローテーションという仕組みがある。これは、長期運用におけるログファイルによるディスク容量圧迫を避けるための機構で、一定期間でログファイル名を変えてバックアップファイルを作成し、バックアップを何世代か保持した上で、最も古いものを捨てるという操作を行うものである。

【０００４】

ログファイルを監視し処理を行う技術に関する従来例には、例えば特開平１０－２９３７０４号公報（特許文献１）、特開平９－１０１９１２号公報（特許文献２）に記載されたものがある。また、ファイルを管理する従来技術として例えば特開２００３－２２２０６号公報（特許文献３）記載のものがある。

【特許文献１】特開平１０－２９３７０４号公報

【特許文献２】特開平９－１０１９１２号公報

【特許文献３】特開２００３－２２２０６号公報

【発明の開示】

【発明が解決しようとする課題】

【０００５】

ログファイルの監視プログラムは、そのログファイルにローテーションが生じていても正しく監視する必要がある。このため、例えば、各監視周期の監視においてオープン中のログファイルの処理を行ったあとローテーションの有無を調べ、ローテーションが生じていればオープンする（すなわち監視対象の）ログファイル名を指定して現在使用のログファイルにオープンし直すことでローテーションに応じたログファイルの監視を可能にしている。ローテーションの有無は、例えばローテーションを行う側の機構（例えばＯＳ：オペレーティングシステム）からの通知により認知することができる。

【０００６】

しかしながら、まれには、監視周期が１周した後の監視においてその間、ローテーションが２度以上生じている場合も考えられる。この場合には、もとのオープン中のログファイルと現在使用のログファイルとは漏れなく監視されるが、その中間で使用されていたログファイルには監視の見落としが発生する。なお、上記の各従来技術では、ログファイル

のローテーションに対応して監視を行うことに言及がない。

【０００７】

本発明は、上記の事情を考慮してなされたもので、ログファイルを監視するためのログファイル監視プログラム、ログファイル監視方法、およびそのようなプログラムまたは方法が適用され得る計算機システムにおいて、ログファイルのローテーションに適切に対応することが可能なログファイル監視プログラム、ログファイル監視方法、および計算機システムを提供することを目的とする。

【課題を解決するための手段】

【０００８】

上記の課題を解決するため、本発明に係るログファイル監視プログラムは、監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶するステップと、監視中でオープン中の前記ログファイルを処理するステップと、前記第１の不変情報が記憶された状態において、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得するステップと、前記記憶された第１の不変情報と前記取得された第２の不変情報とが同一であるか否かを判定するステップと、前記判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更規則に基づいて監視中でオープン中の前記ログファイルに至る世代情報を調査・取得するステップと、前記取得された世代情報に基づいて監視中でオープン中の前記ログファイルより新しい世代のログファイルを順にオープン対象として処理するステップとをコンピュータに実行させることを特徴とする。

【０００９】

すなわち、このプログラムでは、ログファイルが、ローテーションによってそのログファイル名には変更があるが、同時に、ローテーションされても変わらない情報（不変情報）が付与されていることに着目する。監視中でオープン中のログファイルに付与された不変情報を第１の不変情報とし、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報とする。これらが一致すればローテーションは発生していない。一致しなければ何らかのローテーションが生じているので、ログファイル名変更規則に基づいてオープン中のログファイルに遡る世代情報を調査・取得する。世代情報には各世代のログファイルに関する情報が含まれているので、これに基づいて各世代のログファイルを検索すればすべての世代のログファイルについて監視が行き届き処理されることになる。よって、ログファイルのローテーションに適切に対応してログファイルの監視を行うことが可能になる。

【００１０】

また、本発明に係るログファイル監視方法は、監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶するステップと、監視中でオープン中の前記ログファイルを処理するステップと、前記第１の不変情報が記憶された状態において、監視すべきログファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得するステップと、前記記憶された第１の不変情報と前記取得された第２の不変情報とが同一であるか否かを判定するステップと、前記判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更規則に基づいて監視中でオープン中の前記ログファイルに至る世代情報を調査・取得するステップと、前記取得された世代情報に基づいて監視中でオープン中の前記ログファイルより新しい世代のログファイルを順にオープン対象として処理するステップとを具備することを特徴とする。

【００１１】

このログファイル監視方法は、ほぼ上記のログファイル監視プログラムと同様な作用および効果を有する。

【００１２】

また、本発明に係る計算機システムは、監視中でオープン中のログファイルに付与された不変情報を、第１の不変情報として記憶する手段と、監視中でオープン中の前記ログファイルを処理する手段と、前記第１の不変情報が記憶された状態において、監視すべきロ

グファイル名のログファイルに付与された不変情報を第２の不変情報として調査・取得する手段と、前記記憶された第１の不変情報と前記取得された第２の不変情報とが同一であるか否かを判定する手段と、前記判定の結果が同一でない場合に、ログファイルローテーションで適用されるログファイル名変更規則に基づいて監視中でオープン中の前記ログファイルに至る世代情報を調査・取得する手段と、前記取得された世代情報に基づいて監視中でオープン中の前記ログファイルより新しい世代のログファイルを順にオープン対象として処理する手段とを備えたログファイル監視部と、前記ログファイル監視部により監視される前記各ログファイルを記憶するログファイル記憶部とを具備することを特徴とする。

【００１３】

この計算機システムは、上記のログファイル監視プログラムをソフトウエアとして備えた計算機として実現することができる。よって、ここで、上記の各手段は、例えばマイクロプロセッサやメモリ、ハードディスクなどのハードウエアと、これらのハードウエア上で動作するオペレーティングシステム等の基本ソフトウエアや上記のプログラムおよびその他のアプリケーションプログラムなどとによって構成することが可能である。

【発明の効果】

【００１４】

本発明によれば、ログファイルのローテーションに適切に対応してログファイルの監視を行うことが可能になる。

【発明を実施するための最良の形態】

【００１５】

本発明の実施態様として、前記不変情報はｉｎｏｄｅ番号であるとすることができる。ｉｎｏｄｅ番号は、ログファイルのローテーションによっても不変の情報としてよく知られたものである。

【００１６】

また、実施態様として、前記処理は障害発生の検知であるとすることができる。障害発生の検知は、ログファイル監視の用途として広く用いられている。

【００１７】

また、実施態様として、前記ログファイル名変更規則は、前記監視すべきログファイル名に「．ｎ」（ｎは整数で世代が古くなると１ずつ増加）を付加する規則であると仮定する、とすることができる。この規則は、ローテーションによってログファイル名を変更する場合のＯＳ側の規則の典型例であるので、これらにあらかじめ対応するものである。

【００１８】

また、実施態様として、前記世代情報は、ログファイル名と該ログファイル名のログファイルに付与された不変情報とからなる組の情報を前記第２の不変情報から前記第１の不変情報までたどったものである、とすることができる。これにより、監視すべきログファイルを過不足なく調査することができる。

【００１９】

以上を踏まえ、以下では本発明の実施形態を図面を参照しながら説明する。まず、参考例としての形態を図１、図２を参照して説明する。図１は、ログファイルのローテーションを模式的に説明する図（参考例）である。

【００２０】

ある時点において、ｍｅｓｓａｇｅｓの名を有するログファイル１、そのバックアップとしてのログファイル２、さらにそのバックアップとしてのログファイル３が存在するものとする。このようなログファイル１、２、３は、計算機システムにおいて、例えばＯＳやログファイル監視プログラム以外のアプリケーションプログラムがそれらの処理上の必要から保有している。ログファイル１、２、３は、例えばログファイル記憶部としてのハードディスク上に記憶領域が確保されている。

【００２１】

バックアップとしてのログファイル２には、例えばｍｅｓｓａｇｅｓ．１の名、そのバ

ックアップとしてのログファイル３には例えばｍｅｓｓａｇｅｓ．２の名が、これらのログファイルを保有するソフトウエアにより付与されている。ログファイル２、３は、過去のログが収納されたバックアップであり、その時点で使用されている（情報の更新や付加がされる）のはログファイル１のみである。

【００２２】

ログファイルのローテーションがなされると、図示するように、ログファイル１はｍｅｓｓａｇｅｓ．１に自動的に名称が変更されてバックアップとしてのログファイル２Ａとなり、ログファイル２はｍｅｓｓａｇｅｓ．２に自動的に名称が変更されてさらにバックアップとしてのログファイル３Ａとなる。そして、ログファイル３はこの場合破棄される。これにより、新たなログファイルであるがもとの名ｍｅｓｓａｇｅｓを有するログファイル１Ａが新規登場する。

【００２３】

ローテーション後は、ログファイル１Ａのみが使用される（情報の更新や付加がされる）。ログファイルのローテーションの仕組みにより長期運用によるハードディスク上の容量圧迫を避けることができる。なお、この例ではバックアップを２世代とるようにしているが、ＯＳやアプリケーションプログラムにより適宜その増減があり得る。また、ローテーションのタイミングは、例えば１週間というような一定の期間やユーザによる設定があり得る。

【００２４】

ログファイル監視プログラムでは、一般的に、監視すべきログファイル名（この例では「ｍｅｓｓａｇｅｓ」）を登録しその登録のログファイル１をオープンして必要な処理を行う。また、通常はある定められた周期ごとにログファイル１の内容に対して処理を行いその処理以外の期間ではログファイル監視プログラムはスリープ状態となる。一度ログファイル１がオープンされると、そのオープン対象はローテーションにより名が変更（ｍｅｓｓａｇｅｓ→ｍｅｓｓａｇｅｓ．１）になっても当然実質的に同じログファイル２Ａとして継続する。そして、ローテーションが発生した場合には、通常例えばＯＳ側からその情報がもたらされる。ログファイル監視プログラムは、これにより、名称が登録された名称であり新規に登場したログファイル１Ａの監視に移行することができる。

【００２５】

具体的に図２に示す流れ図に基づき説明する。図２は、ログファイル監視プログラムの処理フローを示す流れ図（参考例）である。まず、ログファイル監視プログラムでは、監視するログファイル名（この例では「ｍｅｓｓａｇｅｓ」）を登録する（ステップ２１）。このときほかに監視する周期などが設定可能であればそれを行う。なお、ここで表現として「登録する」の主語がプログラムとなっているが実際にはユーザがキーボードなどのマンマシンインターフェースを使用して登録の指示を行いこれに従いプログラムが動作するということである。ユーザの行う操作については以下でも適宜省略する。

【００２６】

次に、登録されたログファイル名のログファイルをオープンする（ステップ２２）。そして、オープンされたログファイルで監視ポインタを最終の部分に設定する（ステップ２３）。その最終の部分は現在までに存在するログの最終であり、これ以降に付加・更新されるログが監視されることになる。監視ポインタが設定されたら監視周期に従いログ監視プログラムは一定時間スリープとなる（ステップ２４）。

【００２７】

スリープすべき時間が経過したら、次に、オープン中のログファイルの更新を調べる（ステップ２５）。具体的には、例えばログファイルの更新日付を見ることで判断できる。更新がなければスリープ前と監視対象のログファイルに変化はないので次のステップ２６をスキップする。更新がされている場合には、スリープ前に設定されていた監視ポインタの指し示すログ以降に付加されたログについて所定の処理を行う（ステップ２６）。このとき、監視ポインタを処理された最後に進める（同）。「所定の処理」はこのログファイル監視プログラムが有する機能により様々であるが、例えば障害発生の検知などを挙げる

ことができる。

【００２８】

次に、ログファイルのローテーションが発生しているかを調べる（ステップ２７）。これは、例えばＯＳ側から通知された情報に基づいたりＯＳ側に問い合わせることにより可能である。ローテーションが発生していない場合には、現在オープンのログファイルは古くなっていないので次のステップ２８、２９、３０をスキップし、次のスリープ（ステップ２４）に移行する。

【００２９】

ローテーションが発生している場合は、現在オープンのファイルは古くなっている。そこで、そのファイルをクローズし（ステップ２８）、登録されたログファイル名のログファイルをオープンする（ステップ２９）。「登録されたログファイル名」はステップ２１における登録による。そして、オープンされたログファイルについて所定の処理を行い、監視ポインタを最後の部分に設定する（ステップ３０）。これにより、実質的に新たなログファイルがオープン・処理される。この後、次のスリープ（ステップ２４）に移行する。以上のようなフローにより一定周期ごとに所定の処理を行うログファイル監視がなされる。

【００３０】

図３は、図２に示したログファイル監視プログラムの機能を説明する図（参考例）である。図３における、ログファイル１、２、３、１Ａ、２Ａ、３Ａは、それぞれ図１に示した同符号のものに相当する。図３の左側の状態で監視プログラムによる処理がなされ、そのあとローテーションが生じ図３中央の状態になった場合におけるログファイル監視プログラムの処理対象は、上記図２による説明からわかるように、ログファイル２Ａおよびログファイル１Ａのそれぞれドットパターン部である。

【００３１】

今、図３の左側の状態で監視プログラムによる処理がなされ、そのあとローテーションが２度生じ図３の右側の状態になり、次の監視の処理がなされる場合を考える。この場合、図２のステップ２６においてログファイル３ＡＡのドットパターン部について処理がなされる。そのあと図２のステップ３０でログファイル１ＡＡのドットパターン部について処理がなされる。すなわち、中間のバックアップであるログファイル２ＡＡについては所定の処理がなされない。これは監視プログラムとして完全性が欠如しているといえる。

【００３２】

そこで、次に、本発明の一実施形態に係るログファイル監視プログラムについて図４ないし図７を参照して説明する。図７は、本発明の一実施形態に係るログファイル監視プログラムの処理フローを示す流れ図である。

【００３３】

このログファイル監視プログラムでは、まず、監視するログファイル名を登録し、そのログファイルのローテーションによる名変更規則を取得する（ステップ４１）。ログファイル名の登録は図２のステップ２１と同じである。「そのログファイルのローテーションによる名変更規則の取得」は、例えば、図５に示すような名変更規則の情報を監視される側のソフトウエアから得るということである。図５は、ログファイル名が「ｍｅｓｓａｇｅｓ」であれば、１世代前の名は「ｍｅｓｓａｇｅｓ．１」、２世代前の名は「ｍｅｓｓａｇｅｓ．２」であることを、ログファイル名が「ｓｙｓｌｏｇ」であれば、１世代前の名は「ｓｙｓｌｏｇ．ｏｌｄ」、２世代前の名は「ｓｙｓｌｏｇ．ｏｌｄ．ｏｌｄ」であることを情報として取得したことを示している。世代の数は監視される側のソフトウエアの構成に依存する。

【００３４】

次に、登録されたログファイル名のログファイルをオープンし、そのオープンされたログファイルに付与されたｉｎｏｄｅ番号を記憶する（ステップ４２）。ｉｎｏｄｅ番号は、ログファイルに不変に付与されている番号であり、ローテーションが発生しても、ログファイル名のようには変化しない。なお、ｉｎｏｄｅ番号も監視される側のソフトウエア

、例えばＯＳにより付与されるものであるが、ログファイルに不変に付与される情報であればその情報を代わりに利用することもできる。

【００３５】

次に、オープンされたログファイルで監視ポインタを最終の部分に設定する（ステップ４３）。これによりその設定位置以後のログが監視されることになる。監視ポインタが設定されたら監視周期に従いログ監視プログラムは一定時間スリープとなる（ステップ４４）。

【００３６】

スリープすべき時間が経過したら、次に、オープン中のログファイルの更新を調べる（ステップ４５）。具体的には、例えばログファイルの更新日付を見ることで判断できる。更新がなければスリープ前と監視対象のログファイルに変化はないので次のステップ４６をスキップする。更新がされている場合には、スリープ前に設定されていた監視ポインタの指し示すログ以降に付加されたログについて所定の処理を行う（ステップ４６）。このとき、監視ポインタを処理された最後に進める（同）。「所定の処理」は例えば障害発生の検知などを挙げることができる。

【００３７】

次に、登録されたログファイル名のログファイルに付与されたｉｎｏｄｅ番号を取得する（ステップ４７）。「登録されたログファイル名」はステップ４１における登録でのログファイル名である。そして、取得されたｉｎｏｄｅ番号と、ステップ４２で記憶されたｉｎｏｄｅ番号との一致／不一致を調べる（ステップ４８）。これはすなわち、現在オープンのログファイルが古くなっているか否か（ローテーションが発生しているか否か）を調べていることになる。記憶されているｉｎｏｄｅ番号は現在オープンのログファイルのｉｎｏｄｅ番号であり、取得されたｉｎｏｄｅ番号は、登録されたログファイル名のログファイルに付与されたもの、つまり現在使用されているログファイルのｉｎｏｄｅ番号だからである。

【００３８】

両ｉｎｏｄｅ番号が一致の時は、ログファイルのローテーションが発生していないので次のステップ４９から５３をスキップする。そして次のスリープ（ステップ４４）に移行する。

【００３９】

両ｉｎｏｄｅ番号が不一致のときは、ログファイル名変更規則に基づいて、登録されたログファイル名のログファイルからオープン中のログファイルまでの世代情報を遡及的に調査・取得する（ステップ４９）。この世代情報は、ログファイル名とこのログファイル名のログファイルに付与されたｉｎｏｄｅ番号とからなる組の情報を現在のログファイルからオープン中のログファイルに遡ってたどり得るものである。得られる世代情報は、例えば、図７に示すようになる。図７は、図４に示したログファイル監視プログラムにおいて取得された世代情報の例を示す図である。

【００４０】

図７に示すように、まず、登録されたログファイル名「ｍｅｓｓａｇｅｓ」からそのログファイルのｉｎｏｄｅ番号ｘ３が取得できる。次に、ログファイル名変更規則を知得して１世代前のログファイル名「ｍｅｓｓａｇｅｓ．１」がわかっているので、そのログファイル名についてｉｎｏｄｅ番号ｘ２を取得する。ここでｉｎｏｄｅ番号ｘ２が、記憶されたｉｎｏｄｅ番号に一致すれば現在オープンされているログファイルに達したことになる。一致しなければ、さらに、知得されたログファイル名変更規則から前の世代のログファイル名「ｍｅｓｓａｇｅｓ．２」を根拠にそのｉｎｏｄｅ番号ｘ１を取得する。ここで例えばｉｎｏｄｅ番号ｘ１が、記憶されたｉｎｏｄｅ番号に一致したものとすると、この間、ローテーションは２度発生したことになる。

【００４１】

ステップ４９が終了したら、オープン中のログファイルをクローズする（ステップ５０）。そして、その次の世代のログファイルをオープンし、同時にそのｉｎｏｄｅ番号を更

新して記憶する（ステップ５１）。「その次の世代のログファイル」については、取得された世代情報を用いれば、ログファイル名によってもｉｎｏｄｅ番号によっても特定が可能である。ログファイルが新たにオープンしたら、監視に基づく所定の処理を行い、監視ポインタを最後の部分に設定する（ステップ５２）。

【００４２】

次に、オープンされたログファイルのｉｎｏｄｅ番号が、ステップ４７で取得されたｉｎｏｄｅ番号に一致するかを判定する（ステップ５３）。一致の場合は、現在使用中のログファイルがオープンされているので、次のスリープ（ステップ４４）に移行する。一致しない場合には、ステップ５０に戻りそこから同様にオープン対象を更新してステップ５１、５２、５３の処理を行う。以上のようなフローにより一定周期ごとに所定の処理を行うログファイル監視がなされる。

【００４３】

このログファイル監視によれば、監視の１周期の間にローテーションが２度以上発生しても監視の抜けは生じない。これを図６を参照してさらに説明する。図６は、図４に示したログファイル監視プログラムの機能を説明する図である。図６において使用する各引用符号は、図３と同じように付与されている。

【００４４】

図６の左側の状態で監視プログラムによる処理がなされ、そのあとローテーションが生じ図６中央の状態になった場合におけるログファイル監視プログラムの処理対象は、上記図４による説明からわかるように、ログファイル２Ａおよびログファイル１Ａのそれぞれドットパターン部である。これは、世代情報の調査・取得により、オープン対象がログファイル２Ａからログファイル１Ａに変わるからである。ここで、ステップ４７で取得されるｉｎｏｄｅ番号はｘ２であり、それより前に最後に記憶（ステップ５１か４２で記憶）されたｉｎｏｄｅ番号はｘ１である。

【００４５】

また、図６の左側の状態で監視プログラムによる処理がなされ、そのあとローテーションが２度生じ図６の右側の状態になり、次の監視の処理がなされる場合を考える。この場合、図４のステップ４６においてログファイル３ＡＡのドットパターン部について処理がなされる。そのあと図６のステップ５１、５２でログファイル２ＡＡのドットパターン部について処理がなされる。さらにその後、ステップ５３を経て再びステップ５１、５２でログファイル１ＡＡのドットパターン部について処理がなされる。すなわち、中間のバックアップであるログファイル２ＡＡについても所定の処理がなされる。よって、監視プログラムとして監視の抜けが発生しない完全性を有している。ここで、図４のステップ４７で取得されるｉｎｏｄｅ番号はｘ３であり、それよりに前に最後に記憶（ステップ５１か４２で記憶）されたｉｎｏｄｅ番号はｘ１である。

【００４６】

以上説明のように、本実施形態によれば、監視周期間でログファイルのローテーションが複数回発生した場合にも更新・追加されたログを見落とすことなく処理することができる。特に、ログファイル名のローテーションによる変更規則を監視される側のソフトウエアから取得するようにしているので、どのような変更規則であっても対応することができる。ただし、典型的な変更規則であれば、変更規則を仮定しその仮定により前世代のログファイル名が推定できるのでこれによっても同様なはたらきを得ることができる。

【図面の簡単な説明】

【００４７】

【図１】ログファイルのローテーションを模式的に説明する図（参考例）。

【図２】ログファイル監視プログラムの処理フローを示す流れ図（参考例）。

【図３】図２に示したログファイル監視プログラムの機能を説明する図（参考例）。

【図４】本発明の一実施形態に係るログファイル監視プログラムの処理フローを示す流れ図。

【図５】ログファイルローテーションによるログファイル名の変換規則情報の例を示す図

。

【図６】図４に示したログファイル監視プログラムの機能を説明する図。

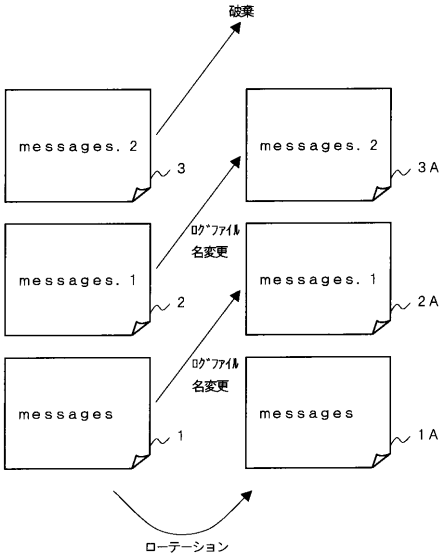
【図７】図４に示したログファイル監視プログラムにおいて取得された世代情報の例を示す図。

【符号の説明】

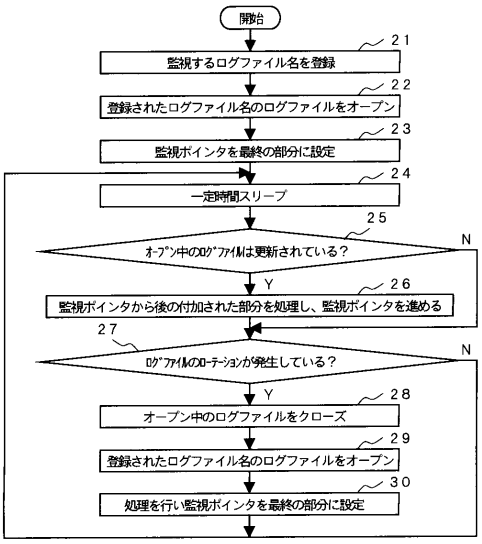
【００４８】

１，１Ａ，１ＡＡ…ログファイル、２，２Ａ，２ＡＡ、３，３Ａ，３ＡＡ…バックアップのログファイル

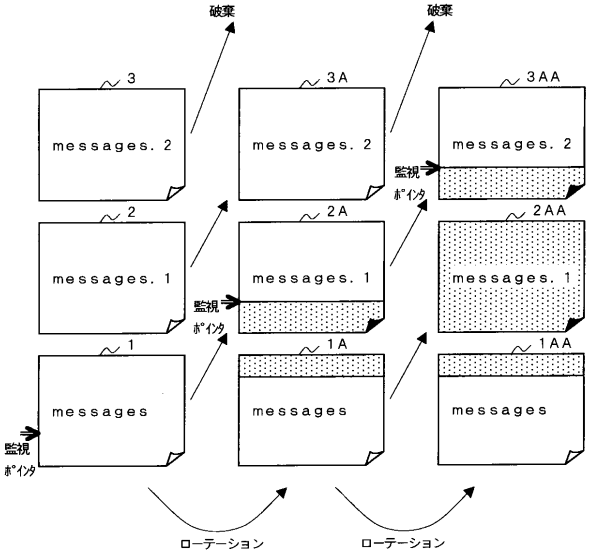
【図１】



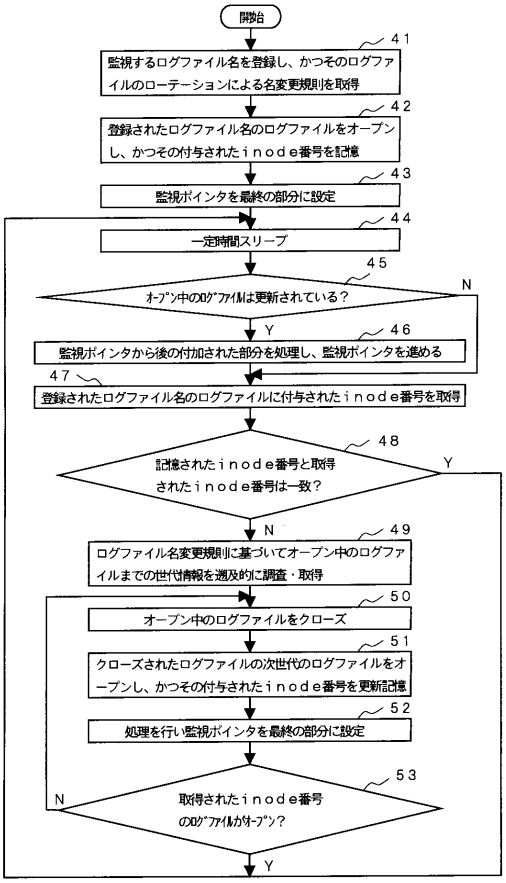
【図２】



【図３】



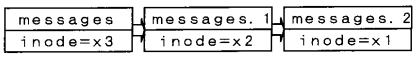
【図４】



【図５】

ログファイル名	1世代前のログファイル名	2世代前のログファイル名	.....
messages	messages. 1	messages. 2	
syslog	syslog. old	syslog. old. old	
⋮	⋮	⋮	

【図７】



【図６】