A Study on Faults and Error Propagation in the Linux Operating System

March 2016

Takeshi Yoshimura

主　論　文　要　旨

報告番号	(甲)乙　第　　号	氏　名	吉村　剛

主論文題目：

A Study on Faults and Error Propagation in the Linux Operating System

(Linux オペレーティングシステムにおけるフォールトおよびエラー伝播に関する研究)

（内容の要旨）

オペレーティングシステムはアプリケーションの信頼性にとって重要である．オペレーティングシステムにおいてフェイラが発生すると全てのアプリケーションのフェイラにつながってしまう．しかし，Android スマートフォンやクラウド基盤，航空管制システムなどの製品で利用されている Linux においても，近年の調査で NULL ポインタ参照のような単純なバグをいまだ発生させていることが明らかになっている．オペレーティングシステムにおけるフェイラを回避する手法の研究は，フォールトの検知とフェイラの回復の方向性に分けられる．フォールトの検知はコードの静的検査やテストなどを利用して運用前に開発者が可能な限りフォールトを修正する手法である．フェイラの回復はソフトウェア若化などを利用して，運用前に修正されなかったフォールトが引き起こすフェイラを回避することや被害を最小限にする手法となる．

フォールトの検知手法やフェイラの回復手法の進展に向けて，本論文は Linux を題材としてフォールトおよびエラーの詳細な調査を行う．これまでフォールトの検知やフェイラの回復手法の改良は様々な研究で取り組まれている．しかし，既存研究はフォールトおよびエラー伝播の全体的な傾向に基づかず，オペレーティングシステムの開発者たちの経験や直感に基づく場当たり的な対策となってしまっている．例えば，NULL ポインタのチェック忘れをするフォールトが多いことを開発者が認識した結果，NULL ポインタのチェック忘れを検査する静的解析が開発されている．フェイラ回復手法はカーネル全体が常に単一エラーによって破壊されることを前提としており，悲観的な方法となっている．

Linux におけるフォールトを調査するため，本論文は 37 万件以上に渡る Linux のパッチに含まれる，英語で記述されたコードの変更説明文を分析する．パッチに含まれるトピックを抽出するため，自然言語処理の手法である Latent Dirichlet Allocation を利用し，抽出されたトピックに基づきパッチを 66 のクラスタに分類する．得られたクラスタが先進的なコード検査につながることを示すため，割り込み処理に関するクラスタの詳細な調査を行い，160 件の割り込みハンドラの解放処理のフォールトを抽出する．抽出したフォールトの知識に基づきコード検査器を開発し，Linux 4.1 において未発見のフォールト 5 件を発見した．

本論文はさらに Linux におけるエラー伝播を調査する．エラー伝播の調査において，新しい概念としてエラー伝播スコープを導入する．エラー伝播スコープはエラー伝播の距離を示す概念である．本論文では２つのスコープである，プロセスローカルエラーおよびカーネルグローバルエラーを導入する．プロセスローカルエラーはカーネル内のプロセスコンテキストに閉じるエラーであり，カーネルグローバルエラーはプロセスコンテキストを超えて伝播するエラーとなる．本論文は実験において 73% のエラーがプロセスローカルであり，カーネル内のプロセスコンテキストを超えて伝播しないことを示す．この結果はフェイルしたプロセスをキルすることで，フェイラから回復する手法の可能性を示している．

本論文の貢献は２つに分かれる．ひとつはフォールトの調査結果により，これまで場当たり的な対策をとってきたために，見逃されてきたフォールトを検知するコード検査器の開発を支援することである．さらに，オペレーティングシステムにおける軽量な回復手法の可能性を示し，今後の調査や研究における課題を明らかにする．

SUMMARY OF Ph.D. DISSERTATION

School	Student Identification Number	SURNAME, First name
Science for Open and Environmental Systems		YOSHIMURA, Takeshi

Title

A Study on Faults and Error Propagation in the Linux Operating System

Abstract

Operating systems are crucial for application reliability. Applications running on an operating system cannot run correctly if the operating system fails. Recent studies reveal that naive faults such as NULL pointer dereferences are still prevalent in the Linux operating system, which is widely used in productions such as Android smartphones, cloud platforms, and air traffic control systems. Existing approaches to prevent operating system failures are twofold: fault detection and failure recovery. Fault detection is the approach to find and fix as many faults as possible before shipping, using techniques such as static analysis and software tests. Failure recovery is the approach to tolerating or mitigating the failures caused by undetected faults, using the techniques such as software rejuvenation.

To advance the state of the art of fault detection and failure recovery for operating systems, this dissertation conducts detailed analysis of faults and error propagations in the Linux operating system. Many efforts have been devoted to improving the quality of fault detection and failure recovery for operating systems. However, existing techniques rely on ad hoc intuitions and experiences of operating system developers without understanding the overall trends in Linux faults and error propagations. For example, if the developers notice that there are many faults in which NULL pointer check is missing, a static code checker is developed to check for missing NULL checks. Failure recovery is pessimistic and assumes the entire kernel is always corrupted by a single error.

To understand faults in Linux, this dissertation analyzes more than 370,000 Linux patches, code modification records for Linux in English. To extract topics in the patches, Latent Dirichlet Allocation, a technique of natural language processing is applied and the patches are classified into 66 clusters based on the extracted topics. To demonstrate the resulting clusters can contain useful information to develop sophisticated code checkers, one cluster is deeply investigated and 160 patches for fixing faults related to interrupt handling are extracted. Based on the knowledge obtained from the extracted patches, a static code analyzer has been developed and detected five unknown faults in Linux 4.1.

This dissertation also investigates error propagations in Linux. To analyze error propagations, a new concept called error propagation scope is proposed. Error propagation scope specifies how far an error can propagate. In the dissertation, two scopes, process-local and kernel-global errors, are introduced. Process-local errors do not propagate beyond process contexts inside the kernel. Kernel-global errors corrupt shared data structures and propagate beyond process contexts. This dissertation shows 73¥% of errors are process-local and do not propagate beyond the in-kernel process contexts in the experiments. This result indicates that there are chances to avoid kernel crashes in Linux by killing a failing process.

The contribution of this dissertation is twofold. First, an analysis of fault reports helps develop new static code checkers that can detect faults overlooked in an ad hoc approach. Second, partial recovery of the Linux operating system deserves further investigation and research to achieve lightweight countermeasures for system failures.