

卒業論文 2019 年度（平成 30 年度）

卒論

慶應義塾大学 環境情報学部

石川 達敬

徳田・村井・楠本・中村・高汐・バンミーター・植原・三次・中澤・武田
合同研究プロジェクト

2020 年 1 月

卒業論文 2019 年度（平成 30 年度）

卒論

論文要旨

アブスト

キーワード

OS

慶應義塾大学 環境情報学部

石川 達敬

Abstract Of Bachelor's Thesis Academic Year 2019

Title

Summary

abst

Keywords

OS

Bachelor of Arts in Environment and Information Studies
Keio University

Tatsunori Ishikawa

目 次

第 1 章	序論	1
1.1	背景	1
1.2	着目する課題	1
1.3	目的	2
1.4	アプローチ	2
1.5	構成	2
第 2 章	関連技術	3
2.1	関連技術 1	3
2.2	関連技術 2	3
第 3 章	設計	4
3.1	マシンの構成	4
3.2	設計	4

图 目 次

表 目 次

第1章 序論

1.1 背景

コンピュータの管理者は、動作中のコンピュータの情報を監視することが必要となる場面がある。例えば `top` コマンドや `ps` コマンドを用いて、プロセスの一覧を得たり、`gdb` コマンドを用いてプロセスをトレースし、プロセスの状態を把握する。

その他にも、様々な手段を用いてログイン中のそのコンピュータ自身の状態を監視・解析する。監視する手段として RDMA NIC が登場したことをのべる。

1.2 着目する課題

コンピュータの状態は、コンピュータ内部におけるレジスタの値および、内部から参照できる仮想アドレス空間上に保持されている。

例えばコンテキストスイッチでは、`task_struct` 構造体から辿れる退避されたメンバから値を取り出すことでプロセス空間および状態の復元を行なっている。

`task_struct` 構造体をはじめとして、*Linux* カーネルの変数や型、関数は、様々なアーキテクチャやカーネルコンフィグに対応するため、マクロによって分岐されている。この分岐が確定するのは、*Linux* カーネルをビルドするときであり、構造体のメンバへのアクセス、関数のアドレスなどはコンパイラが保証している。

実際のカーネルのバイナリは、`vmlinux` としてコンパイルされた後、`strip` され `bzImage` となる。ユーザーが作成したカーネルモジュールなどで関数を呼び出す際は、シンボルとアドレスの変換表である `/boot/System.map` を参照し、実際の

背景で述べた RDMA NIC を用いた解析手法では、メモリの物理アドレスを指定し、逐次的に値を取得し外部から復元していくが、CPU レジスタの現在の値は直接知ることができないため、例えばプロセスの一覧を取得したい場合は、コンテキストスイッチ時に退避された値を辿っていく必要がある。しかし、上述の通り `task_struct` はビルドされた際のカーネルコンフィグによって、どのメンバが先頭アドレスからどのオフセットに保持されているかは変動する。

Linux カーネルのバージョン、`System.map` の情報および `config` の情報を知らなければ、メモリのみから正しくコンテキストを復元していくことはできない。

`libtlp` の論文（もうちょっとちゃんと書く）で紹介されている `process-list.c` は `System.map` を引数として渡し、`init_task` の行を `read` することで `init_task` の仮想アドレスを得ている。また、カ

ーネルコンフィグに依存するマクロの値や，使用する関数などもハードコーディングされており，あの論文の環境以外で動かすことが容易ではない．

1.3 目的

そこで本研究の目的として，問題の章であげた 3 つの情報，Linux カーネルのバージョン，System.map の情報およびカーネルコンフィグのうち，System.map およびカーネルコンフィグの情報を RDMA NIC を用いて復元する．また，Linux カーネルのバージョンを知ることさえできれば，どのようなカーネルコンフィグを持つコンピュータに対しても，プロセスリストの一覧を取得できることを実証する．

1.4 アプローチ

目的を達成するための手段として，本研究ではいくつかの工程を踏むこととする．

1.5 構成

構成

第2章 関連技術

ああああ

2.1 関連技術 1

説明

2.2 関連技術 2

説明

第3章 設計

アブスト

3.1 マシンの構成

あああ

3.2 設計

あああ