

1 はじめに

現在、我々はカーネルレベル命令トレーサ『鶴』と、鶴の出力したデータを解析するツール群『鶴Tools』を開発している。鶴は、オペレーティングシステム(以下OS)とアプリケーションの動作を1命令毎にトレースし記録する。鶴は、1命令当たり平均15.5byteのレコードを静的メモリに出力する。したがって、十分な静的メモリが確保できない場合には、2次記憶装置の利用が不可欠となる。しかし、鶴はカーネルレベルにおいて割込み禁止状態で動作するため、OSで提供されている機能を用いてトレースデータを2次記憶装置へ出力できない。本論文では上記の問題を解決するために開発したログ機構について述べる。本ログ機構は、特定のデバイスに非依存であり、様々なデバイスにデータを出力することができるという特徴を持つ。本論文では、特に、ログ出力対象デバイスをIDE-HDDとEthernetとした場合の実装について述べる。

2 ログ機構の構成と設計

本ログ機構は、デバイス非依存のデータ出力機構をユーザに提供することを目的とする。

本ログ機構は、以下の3つの部からなる。

- (1) 初期設定部
- (2) バッファ蓄積部
- (3) デバイス出力部

全体構成を図1に示す。(1)(2)はデバイス非依存であり、(3)はデバイス依存である。このような構成にすることにより、デバイス非依存である(1)(2)が様々なデバイスを同一インターフェースで使用することができる。次に設計とインターフェースについて述べる。

Development of Logging Mechanism for Instrucion Tracer
Syoichiro KOMIYAMA, Hiroyuki MORIMOTO, Koichi MOURI,
Yasufumi YOSHIZAWA
Faculty of Technology, Tokyo University of Agriculture and
Technology

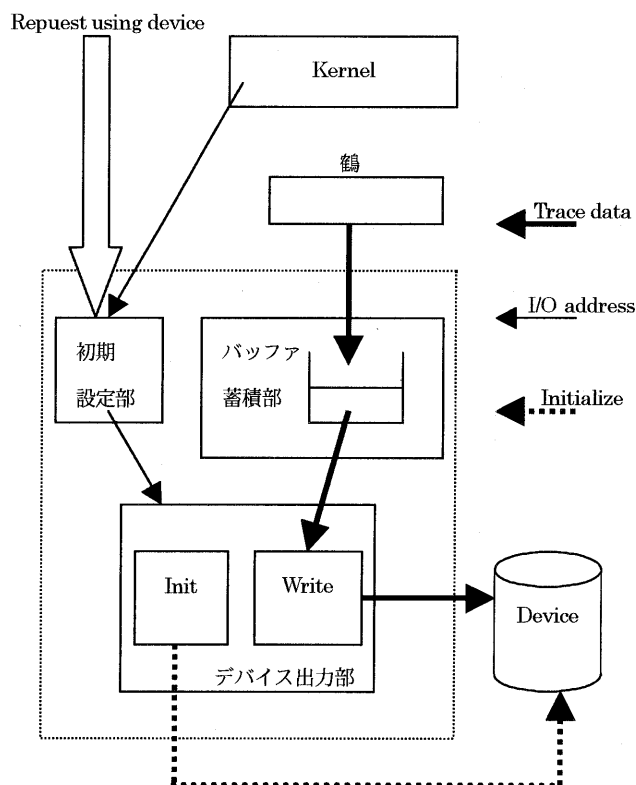


図1: ログ機構の全体構成

2.1 初期設定部

ユーザから使用要求のあったログ対象デバイス名をカーネルに問い合わせ、そのI/Oアドレスを得る。その値をデバイス出力部へ渡す処理を行う。

2.2 バッファ蓄積部

鶴は、1命令当たり平均15.5byteのレコードを静的メモリに出力する。そのため、1つのレコードを出力する度にデバイスへ書出しを行ったのでは、入出力待ち時間が大きくなり、鶴の処理効率が低下する。したがって、トレースデータのバッファリングが必要になる。バッファ蓄積部では、トレースデータへのポインタとそのサイズへのポインタを鶴から受け取り、バッファへデータを蓄積する。バッファサイズは、鶴のオーバヘッドを軽減するために最適にする必要がある。現在は64Kとしている。