

ドライバプログラムの効率的な構成制御法

滝口真一[†] 田端利宏[‡] 乃村能成[‡] 谷口秀夫[‡]

岡山大学工学部[†] 岡山大学大学院自然科学研究科[‡]

1. はじめに

入出力機器の増加とともに、オペレーティングシステムは、多くのドライバプログラム(ドライバ)を保有する。このため、占有メモリ量の増加、および立ち上がり時間の長大化といった問題が生じている。そこで、本稿では、ドライバを効率的に構成する制御法について述べる。具体的には、既存 OS の問題点を示し、ドライバの読込、初期化、および終了を制御する方式を提案する。

2. 既存 OS の問題点

2.1 既存ドライバの構成

利用環境に合わせて、必要最小限のドライバを保有する OS を構成するには、適切なドライバの組み込みが必要である。しかし、カーネル構築は種々の設定を必要とし、専門的な知識を必要とする。また、設定工数も小さくない。このため、多くの場合、標準的な構成の OS を利用する。

したがって、利用者が利用する環境においては、不要なドライバを保有する OS を利用することになってしまう。この結果、ドライバによる占有メモリ量は大きくなり、またドライバの初期化処理に伴い立ち上がり時間は長くなっている。

2.2 問題の定量化

2.2.1 環境

ドライバが占有メモリ量と立ち上がり時間を与える影響を明らかにするため、FreeBSD 4.3 を用いて評価を行った。利用した計算機ハードウェアは、CPU : Intel Pentium4 2.8GHz, メモリ : 512MB である。OS として、標準カーネルと小カーネルを用意した。標準カーネルは、多くの計算機環境で動作するよう構成されており、ドライバを含め 175 個のモジュールが組み込まれている。小カーネルは、不要なドライバを省いたカーネルであり、55 個のモジュールが組み込まれている。

2.2.2 占有メモリ量

コマンド kldstat を利用してログイン直後の占

有メモリ量を測定した。結果を表 1 に示す。表 1 より、標準カーネルは、小カーネルに比べ、約 1.43MB 占有メモリ量が多い。この値は、標準カーネルの 41.32%, 小カーネルの 70.42%に相当し、ドライバが OS の利用するメモリ量に占める割合が非常に大きいことが分かる。

表 1 占有メモリ量

	サイズ(Byte)
標準カーネル	3,470,924
小カーネル	2,036,636

2.2.3 立ち上がり時間

CPU のタイムスタンプカウント値を取得する rdtsc 命令を用いて、電源投入から login 起動までの時間(立ち上がり時間)を測定した。測定結果は、測定回数 10 回の平均である。結果を図 1 に示す。

図 1 より、標準カーネルは、小カーネルに比べ、非 ISA ドライバと ISA ドライバの設定時間が大きい。ここで、立ち上がり時間とドライバ初期化(ドライバ登録、非 ISA ドライバ設定、ISA ドライバ設定)時間の関係を表 2 に示す。表 2 より、標準カーネルは立ち上がり時間のうちドライバ初期化に 24.04%を占めているのに対し、小カーネルは 5.06%を占めている。つまり、不要なドライバの初期化処理に使用される時間は大きい。

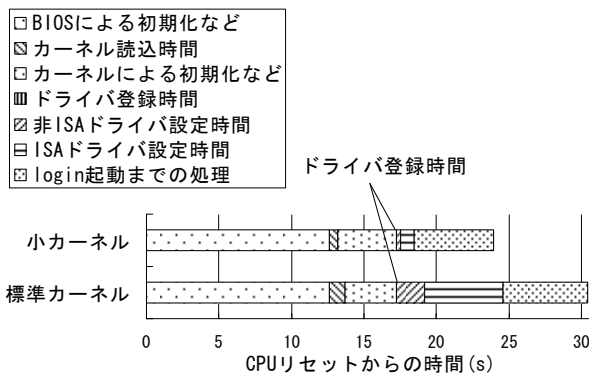


図 1 立ち上がり時間

表 2 立ち上がり時間とドライバ初期化時間の関係

	立ち上がり時間	ドライバ初期化時間(比率)
標準カーネル	30.42s	7.31s (24.04%)
小カーネル	23.94s	1.21s (5.06%)

Efficient control of Drivers construction
Shinichi Takiguchi[†], Toshihiro Tabata[‡], Yoshinari Nomura[‡] and Hideo Taniguchi[‡]
[†] Faculty of Engineering, Okayama University
[‡] Graduate School of Natural Science and Technology, Okayama University

3. 効率的な構成制御

3.1 基本方針

ドライバを効率的に構成する制御法における基本方針は、「必要なときに必要なドライバのみを動作させる」ことである。

この基本方針を実現するにあたり、「入出力機器(デバイス)」、「ドライバ」、および「APからの利用状況」の関係を明確にする必要がある。また、ドライバの動作履歴情報を利用して、効率的なドライバの起動(読込と初期化)と終了を制御する必要がある。

以降では、上記2項目について詳しく述べる。

3.2 ドライバ制御状態

デバイス、ドライバ、および AP からの利用状況の組み合わせに基づく状態をドライバ制御状態と定義し、議論する。

デバイスは、存在の有無の2状態を持つ。ドライバは、非読込、読込済み、初期化済みの3つの状態に分けられる。利用状況は、非利用と利用中の2つの状態に分けられる。以上の組み合わせにより、ドライバ制御状態は12通り考えられ、実現可能な観点から表3に示す6つの状態に絞る。この6つの状態間の遷移を図2に示す。

各状態は、デバイスの取付けと取外し、ドライバの読込と初期化と終了、利用状況の利用要求と利用終了により遷移する。

表3 ドライバ制御状態

	デバイス	ドライバ	利用状態
(1)無し	無	非読込	非利用
(2)装着待ち	無	読込済み	非利用
(3)装着のみ	有	非読込	非利用
(4)初期化待ち	有	読込済み	非利用
(5)利用待ち	有	初期化済み	非利用
(6)利用中	有	初期化済み	利用中

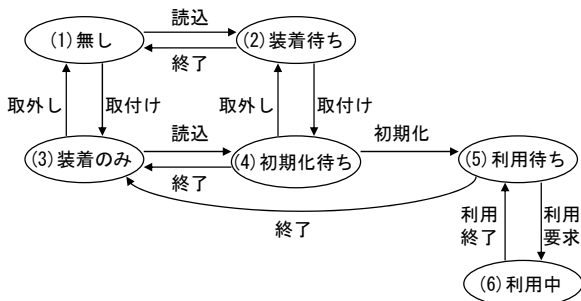


図2 ドライバ制御状態の遷移

3.3 APからの利用に向けた制御

ドライバの読込、初期化と終了を制御する方法として、以下の5つを考える。

(事前読込)：デバイス検出前にドライバを読み込み、デバイス検出時にドライバを初期化する。

(即時起動)：デバイス検出時にドライバを読み込み、初期化する。

(遅延起動)：デバイス検出時にドライバを読み込まず、その後の空き時間に読込と初期化を行う。

(利用要求時起動)：APから利用要求が来た時にドライバを読み込み、初期化する。

(自動削除)：利用終了後にドライバを終了させる。

デバイスの利用回数や利用周期といった情報をドライバ動作履歴情報として収集し、制御を行う。例えば、ドライバ動作履歴情報を用い、利用度の程度にあわせ、利用度が高い順に、(事前読込)、(即時起動)、(遅延起動)、(利用要求時起動)を行うように制御する。また、利用度が低く、かつ利用周期が長いデバイスについては、(自動削除)を行う。これらにより、カーネルが占有するメモリ量を抑制し、立ち上がり時間の短縮化を図る。

これらの制御方法の開始契機を図3に示す。ドライバはOS起動から利用要求までの間に4つの契機のいずれかで読込と初期化を行う。利用終了後、ドライバ動作履歴情報に基づいて(自動削除)を行う。PnPデバイスは[取り付け]と[取外し]が、利用者により行われる。

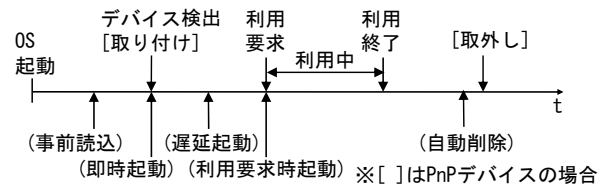


図3 制御の開始契機

4. おわりに

既存OSは、不要なドライバを含む。このため、標準カーネルでは、不要なドライバが利用メモリの約4割を占有し、立ち上がり時間が2割以上遅くなっている。そこで、ドライバの効率的な構成制御法を述べた。具体的には、ドライバ動作履歴情報を利用し、ドライバの読込、初期化、終了を制御する。今後は、詳細検討を行い、**AnT** オペレーティングシステム[1]に実現する予定である。

謝辞 本研究の一部は、科学研究費補助金 基盤研究(B)「適応性と頑健性を有する基盤ソフトウェアのカーネル開発」(課題番号:18300010)による。

参考文献

[1]谷口秀夫, 乃村能成, 田端利宏, 安達俊光, 野村裕佑, 梅本昌典, 仁科匡人, “適応性と堅牢性をあわせ持つ**AnT** オペレーティングシステム,” 情報処理学会研究報告, 2006-OS-103, Vol. 2006, No. 86, pp. 71-78, Jul. 2006.