

Linux の制御システムへの応用に関する研究

吉 田 秀 之*

[要 旨]

本調査研究は、Linux を制御システムのOSとして利用する方法について研究し、制御システムの効率的な開発を図ることを目的とする。

そこで、Linux でGUIを用いたプログラミング、制御機器を利用するためのI/O制御プログラミング、デバイスドライバの開発手順、及び制御の周期性について調査、研究を行った。今回はその中でも制御システムでは重要となる制御の周期性に関して報告する。

結果として、リアルタイム系のLinuxを用いることなく、一般的なLinuxでも比較的簡単に一定の周期性を得ることが確認できた。

1. 緒 言

パソコン等の制御機器から外部機器を制御する場合、基本的にはリアルタイム処理が行えるOSが要求されるが、タイミング的に非常にクリティカルなものを除いては、通常、MS-DOSかMS-WindowsがOSとして利用されている。

しかし、MS-DOSは高度なGUIを持たない、MS-Windowsは制御プログラミングが複雑である等それぞれ一長一短があり、制御システムを開発するOSとして適しているとはいえない。

そこで、オープンソースソフトウェアの一つであるLinuxを制御システムのOSとして利用する方法について研究することで、制御システムの効率的な開発を図ることを目的とする。

今回は、以下の項目について調査、研究を実施した。

- (1) GUIプログラミング手法の検討
- (2) I/O制御手法の確立
- (3) デバイスドライバ開発手法の確立
- (4) 周期性の確立

この(1)~(3)は、主に制御システムのハードウェアを制御するためのプログラミング手法であり、文献調査を行うことで手法を習得した。

(4)は、何らかの制御を行う場合、一定の時間間隔毎に制御が行える必要があり、言い換えれば任意の時刻に制御できなければいけない。これには周期性を持った制御能力が要求される。Linuxには、制御用にリアルタイム処理が可能なLinuxが存在するが、一般的ではなく使いこなすのも容易ではない。

そこで、リアルタイム系のLinuxを用いることなく一般的なLinuxで、どの程度の周期性を持ち備えており、また、どのような方法で周期性を確立させるかについて検討し、次のような内容で実験を行った。

2. 実験方法

実験に用いた環境は次のとおり。

O S : RedHat7J (Kernel バージョン2.2)

C P U : PentiumII 450MHz

メモリ : 128MB

また周期性の測定は、I/Oポート (パラレル

* 企画総務課 主任

ポート) に信号を出力し、再び同じポートに信号を出力するまでの時間を周期時間としてソフトウェアで計測した。ソフトウェアで時間を計算する方法は、Pentium 固有の命令である RDTSC (Read Time Stamp Counter) を用いた。

これらをC言語で作成し、X-Windows上で動作させ測定を行った。

3. 結果及び考察

まず、一切処理を施さない通常の使用状態における周期性を確認するため、100,000回連続して周期時間を測定した。その結果を図1に示す。

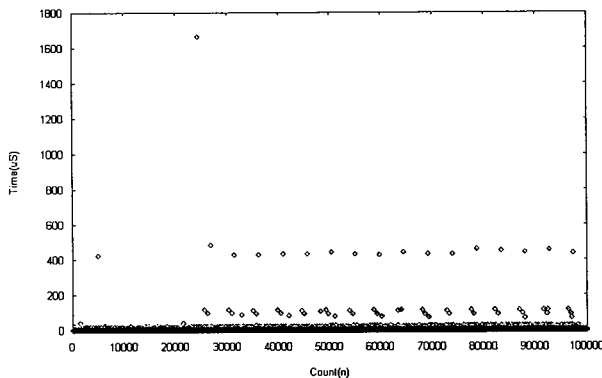


図1 通常状態の周期時間

この場合の最大周期時間は1665 μ S、最小周期時間は1.5 μ S と一定では無く、また、この値は実験毎に異なったものとなるため、制御システムに必要とされる任意の時刻に制御することは不可能である。

次に、Linuxのカーネルタイマを利用して周期時間を一定にする方法を用いて同様に測定した。その結果を図2に示す。

この方法の場合、周期時間そのものは長くなっているが最大周期時間が10.05mS、最小周期時間が10.00mS、平均10.02mS とほぼ一定の周期時間を再現することが可能である。この周期時間は、Linuxのプロセス管理におけるスケジューリン

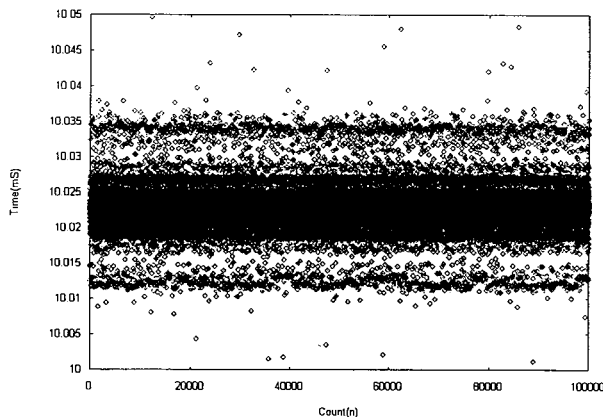


図2 カーネルタイマを利用した場合の周期時間

グのタイミングで発生するものである。ただし、タイマ割り込みを利用している関係で、制御プログラムはカーネルモードでのプログラミングが要求されるので注意が必要である。

もう一つの方法として、Linuxのプロセス管理を応用し、常に優先度を高めることで周期時間を一定にする方法を用いて測定した。その結果を図3に示す。

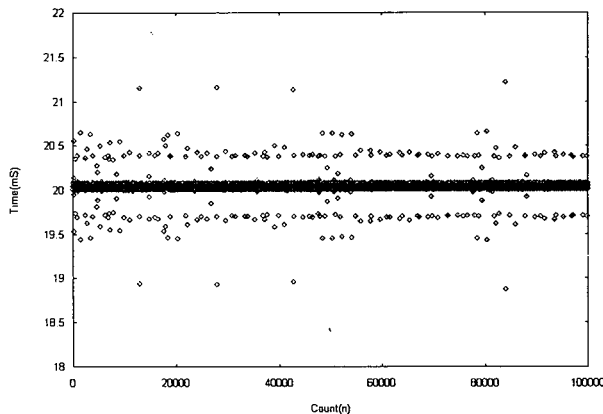


図3 プロセス管理を応用した場合の周期時間

この場合の最大周期時間は21.22mS、最小周期時間は18.87mS、平均20.05mSであり、この方法でもほぼ一定の周期時間を再現することが可能である。さらに、前述の方法のようにカーネルモードではなくユーザモードで動作するため、煩わしいカーネルモードのプログラミングを行わず制御プログラムの作成が可能である。

4. 結 言

本調査研究では、Linux で GUI を用いたプログラミング、制御機器を利用するための I/O 制御プログラミング、デバイスドライバの開発手順について調査をしその手法を習得した。

また、Linux のプロセス管理について調査を行ったところ、スケジューリングは 10mS 毎に行われていることがわかった。このプロセス管理を応用することで、制御可能な周期時間を 10mS (または 20mS) 以上 10mS 単位で比較的簡単に得られることが実験を通して判明した。なお、この周期時間はカーネルを再構築することで変更することができ、さらに周期時間を短くすることが可

能である。

すなわち、この周期時間で制御が可能な制御システムには、OS として Linux を適用することが可能である。

(参考文献)

- ALESSANDRO RUBINI : Linux デバイスドライバ (1998)
- 山岡賢一 : 技術者のための UNIX 系 OS 入門、P42 (2000)
- 熊谷正朗 : Linux Japan、4 巻 - 7 号、P40 (2001)