

多関節型ロボットとパラレルメカニズムの 協調制御による組立作業

藤原基芳*, 中北賢司*

Assembly Task with Cooperation of Articulated Robot and Parallel Kinematic Machine

by Motoyoshi FUJIWARA and Kenji NAKAKITA

It is said that the assembly of many kinds and small lots is difficult in the manufacturing industry. Large lots assembly can be automated with automated machine. When we assemble many kinds and small lots, we assemble many kinds one after the other. When we use automated machine for this task, we must change setup when the lot changes. So we can assemble them faster with hand than with automated machine. For the study of small lots assembly, we introduced robot system. It uses PC/AT platform for control computer. Windows is usually used for PC/AT platform, but we can't do realtime control with that. In this report, we looked over Linux based realtime OS, installed ART-Linux, and wrote a realtime program to get F/T sensor data.

Key Words: assembly, realtime OS, PC/AT platform, Linux

1. はじめに

製造業において、少量生産品の組立作業は自動化が難しいと言われている。大量生産品の組立作業は、専用の組立機を作ることにより自動化できるが、少量生産品の組立作業においては、専用の自動機を製作したとしても、段取り替えが多く、効率が上がらないからである。

少量生産品の組立作業の研究を行うため、平成12年度に研究用ロボットシステムを導入した。このシステムは、制御用コンピュータとして、PC/AT互換機を用いている。PC/AT互換機には、通常Windows系OSが用いられている。しかし、Windows系OSには、ロボットの制御に必要な実時間処理ができない、という問題点がある。

PC/AT互換機上で動作する実時間処理OSには様々

なものがあるが、近年フリーのOSであるLinuxを改造した実時間処理OSが注目されている¹⁾。

本報告では、PC/AT互換機上で動作する、Linuxを用いた実時間処理OSについて説明する。

2. Linuxを用いたリアルタイムOS

2. 1 リアルタイムOSについて

リアルタイム処理とは、他のデバイスからの入力信号や、プログラムからの要求に対して、即座に（リアルタイムに）これを処理する方式である。

制御システムなどでは、一定時間内に処理を確実に終了しなければならない場合がある。このようなリアルタイム性を保証し、許容される時間内に処理の完了を保証する処理方式がリアルタイム処理である²⁾。

PC/AT互換機で通常用いられているWindows系のOSではこのような処理はできない。リアルタイム処理を行う場合、リアルタイムOSと呼ばれるものを使う。

* 機械情報電子グループ

PC/AT互換機を用いたリアルタイムOSとして、VxWorks(Tornado)/LynxOS/Linuxを用いたリアルタイムOS, 等がある¹⁾。この中で、Linuxを用いたリアルタイムOSは以下の点で優れている。

- ・Linuxのもつ豊富な市販ハードウェアへのデバイスドライバと豊富な汎用OSの機能を用いることができる。
- ・無料で入手できる。

Linuxおよび改造を加えたLinuxを用いた実時間OSには以下2. 2～2. 4のようなものがある。

2. 2 汎用Linux^{2),4)}

汎用Linuxのタイマ割り込み周期を短くし、周期実行させたいプロセスの優先度を高くする方法である。以下の特徴がある。

- ・機能に一切変更を加えないLinuxを使用するため、安定性が高く、環境構築も容易である。
- ・かなりの確率でプログラムの周期実行が可能であるが、保証はできない。
- ・プログラムを実行するときに様々な小細工が必要で煩わしい。

2. 3 RT-Linux¹⁾

Linuxのリアルタイム拡張である。Linuxカーネルは最も優先度の低いリアルタイムプロセスとしてスケジューラされる。以下の特徴がある。

- ・開発体制が整っており、運用実績も多いので、OSのバグは少ないと考えられる。
- ・非実時間プロセスからはLinuxの豊富なデバイスドライバが使用可能であるが、実時間プロセスは非実時間プロセスと通信することで間接的にのみLinuxのデバイスドライバを使用することができる。したがって、通常のデバイスドライバのコード中で長すぎる割り込み禁止区間を持っていた場合には、実時間プロセスが動作を停止する可能性があり、これを防ぐためにはリアルタイム処理用のデバイスドライバを開発し直さなければならない。
- ・133[MHz]を超えたPentiumではCPUやOSのスケジューリング性能よりも先にCPUバスの処理能力が限界となる。すなわち、ハードウェアの限界に近い周期でのリアルタイム処理が可能。
- ・制御ルーチンからメモリも含めた全てのハードウェアに対して直接操作が可能である。それゆえ制御を容易に行うことができるのであるが、プログラムにミスがあったり、誤ったメモリを操作すると、OS

にダメージを与え、コンピュータが停止に至る可能性がある。⁴⁾

- ・固定優先度によるスケジューリング。ただし、OSでは優先度逆転に対処していない。ユーザープログラムでの対処が必要。

2. 4 ART-Linux^{5),6)}

Linuxのリアルタイム拡張である。以下の特徴がある。

- ・石綿氏一人で開発しているので、バグフィックス、新しいバージョンのLinuxカーネルへの対処といった点で不安(2002年4月現在、Linuxカーネルは2.4系列が出ているのに対し、ART-LinuxパッチのPC/AT互換機用のものはカーネル2.2.19対応のものが最新)。
- ・デハイスドライバはソース互換。ソースコードをART-Linux上で再コンパイルすることによりリアルタイムプロセスからLinuxの豊富なデバイスドライバを利用可能。
- ・プログラムはメモリが保護された状態で動作する。すなわち、あるプログラムがOSの管理領域や他のプログラムの動作領域に操作を行うと、保護が働き、プログラムは停止させられる。そのため、プログラムの誤りではOSの停止に至ることは非常にまれであり、安全である。
- ・固定優先度によるスケジューリング。また、優先度継承により優先度逆転に対処。
- ・新しいシステムコールが少なく、プログラムを記述しやすい。

2. 5 Linuxを用いたリアルタイムOSについて

以上をまとめると、表1のようになる。この中でART-Linuxは開発体制については不安があるが、ユーザー側の使いやすさ、ユーザー側のミスへのOS側の対処、といった点で優れている。したがって、著者らはART-Linuxを採用することにした。

3. リアルタイム処理の実行³⁾

3. 1 ART-Linuxのインストール

WindowsがインストールされているパソコンへのART-Linuxのインストールは、以下のような手順になる。

- 1 Windowsのdefragを実行。
- 2 FIPSでハードディスクのパーティションを分割。
- 3 汎用Linuxをインストール。

表1 Linuxおよび改造を加えたLinuxの比較

	Linux	RT-Linux	ART-Linux
OSの開発体制	◎	○	×
インストールしやすさ	○	△	△
OSの安定性	◎	○	△
割り込み遅延時間	△	◎	○
デバイスドライバのLinuxとの互換性	◎	○	◎
ソースコードの記述しやすさ	△	○	◎
メモリ保護	◎	×	○
優先度逆転の抑制	×	×	○
実行時の使いやすさ	△	◎	◎

4 Linuxのカーネルバージョンに合致したART-Linuxパッチをあて、デバイスドライバ及びカーネルを再コンパイル。

3. 2 リアルタイム処理プログラムの実行

図1は、研究用ロボットシステムのパラレルメカニズムの手先に取り付けた力覚センサのデータを取り込むC言語プログラムである。OSはLinuxである。

このプログラムのコンパイル及び実行手順は以下のとおりである。

手順1

- 1 "cc art_ifs.c -0"と入力し、コンパイル。
- 2 "su"コマンドを実行し、rootユーザーになる。
- 3 "./a.out"と入力し、実行。
- 4 何かキーを押すまで力覚センサデータを取り込む。

しかし、このプログラムはデータ取り込み周期が不定である。同じパソコンで他のプログラムを実行すると、取り込み周期が大きく乱れる。

そこで、以下のようにする。

- 1 図1の"//"を全て取り払う。
- 2 "cc art_ifs.c -0 (フォルダ名)/art_syscalls.o"と入力し、コンパイル。

3 以下、手順1に同じ。

これだけの変更で、500 μ secごとに力覚センサデータを取り込むようになる。

4. まとめ

1 Linuxを基にしたリアルタイムOSについて調べた。

汎用Linuxを用いる方法は、OSの信頼性は高いが、プログラムの周期実行を保証できず、プログラムを実行するときに様々な小細工が必要である。

RT-Linuxは、OSの信頼性が高く、ハードウェアの限界に近い周期でのリアルタイム制御が可能であるが、優先度逆転への対処とメモリ保護がされていない。

ART-Linuxはプログラムの周期実行が保証されており、また優先度逆転への対処やメモリ保護がなされているが、OSの開発体制に不安がある。

2 上記の点を考慮し、研究用のロボットシステムにART-Linuxをインストールした。これにより、力覚センサデータ取り込み用プログラムに簡単な変更を加えるだけで、一定周期でデータの取り込みを行えるようになった。

参考文献

- 1) 加賀美聡: “ロボット研究のためのPC/AT互換機上のリアルタイムOS”. 日本ロボット学会誌, 16(8), p1036-1041(1998)
- 2) “ASCII24 - アスキー デジタル用語辞典 - リアルタイム処理”. <http://yougo.ascii24.com/gh/21/002108.html>
- 3) 石綿陽一: “ART-Linuxにおける実時間処理とLinux kernel 2.2シリーズ対応版について”. <http://www.movingeye.co.jp/~you1/art-linux/rlc200101/text.html>
- 4) 熊谷正朗: “非RT-Linuxによるロボット制御”. <http://www.mechatronics.mech.tohoku.ac.jp/~kumagai/linux/>
- 5) 熊谷正朗ほか: “汎用Linuxによるロボット制御システムの開発”. 日本ロボット学会誌, 20(2), p157-163(2002)
- 6) 石綿陽一ほか: “高度な実時間処理機能を持つLinuxの開発”. 第16回日本ロボット学会学術講演会予稿集, 1, p355-356 (1998)

```
#include <stdio.h>
#include <asm/io.h>
#include <unistd.h>
#include <sys/io.h>
#include <linux/isicom.h>
#include <sys/time.h>
#include <sys/types.h>
#include <unistd.h>

#include <stdlib.h>
//#include <linux/art_task.h>

#define baseAddr      0x3E0

int mykbhit(){
    fd_set rfds;
    struct timeval tv;
    int retval;

    FD_ZERO(&rfds);
    FD_SET (0, &rfds);

    tv.tv_sec=0;
    tv.tv_usec=0;

    retval=select(1, &rfds, NULL, NULL, &tv);
    if(retval){
        return 1;
    }else{
        return 0;
    }
}

int getdata(int addr)
{
    outw(addr, baseAddr);
    return inw(baseAddr+2);
}

int putdata(int addr, int data)
{
    outw(addr, baseAddr);
    outw(addr, baseAddr+2);
}

main(){
    int i;
    int status=0;
    ioperm(0x000, 0x03ff, 1);

    putdata(0x00e7, 0x0800);
    // art_enter(ART_PRIO_MAX, ART_TASK_PERIODIC, 500);
    while(!status){
        // art_wait();
        getdata(0xa0)/16384*(float)getdata(0x80);
        status=mykbhit();
    }
    // art_exit();
}
```

図1 力覚センサのデータ取り込み用C言語プログラム