

## Linuxを用いたマイクロコンピュータの開発システムの構築

堀場隆広\*1 安田篤司\*2 盛田耕作\*1

Establishment of a Microcomputer Development System Using Linux

Takahiro HORIBA, Atsushi YASUDA and Kosaku MORITA

本研究ではLinuxをOS(Operating System)として動作するパソコンを組み立て、インターネット上に公開されているオープンソースを用いて、安価なマイクロコンピュータの開発システムを構築した。構築した開発システムのソフトウェアは、クロスCコンパイラとマイクロコンピュータの転送プログラムから成っている。この開発システムを用いて簡単なプログラムを作成し、市販のマイクロコンピュータボードにプログラムを転送してその動作を確認した。

## 1. はじめに

オープンソースは、プログラムのソースコードを無償で公開し、多くの人にプログラムを利用してもらい、バグの少ない完成度の高いプログラムを短期間で作成するという開発手法である。また、プログラムのソースコードが公開されているため、だれにでもプログラムの入手、変更、再配布ができる。インターネットの急速な普及により、オープンソースの考え方は、世界中に広まり、インターネットを通じて、プログラムの公開が容易にできるようになった。代表的なオープンソースとして、フィンランドのLinus Torvalds氏によって開発されたUNIX互換のOSであるLinuxとアメリカのRichard Stallman氏が中心となって設立されたFSF(Free Software Foundation, Inc.)において公開されているオープンソース群がある。

本研究では、LinuxとFSFにおいて公開しているオープンソースを用いてマイクロコンピュータの開発システムを構築したので報告する。

## 2. 開発システム

開発システムの構成を図1に示す。開発システムは、市販のマザーボード等のパソコン用部品から開発システム(Linux)の構築に適したしたものを選択・組み合わせたものである。このパソコンにマイクロコンピュータ(以後、マイコンと呼ぶ。)のプログラムを開発するためのプログラム(クロスコンパイラ)と組込用のマイコンにシリアルインタフェースを用いて転送するた

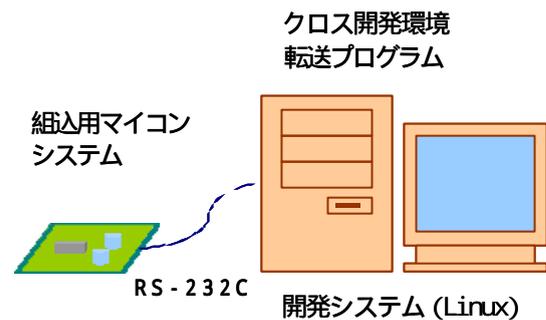


図1 開発システム

めにプログラム(以後、転送プログラムと呼ぶ。)を入れて、開発システムとした。

このシステムを用いたマイクロコンピュータのプログラムの開発手順は、以下のとおりである。まず、開発システムに入っているテキストエディタを用いて、マイクロコンピュータのソースプログラムを作成する。次にできたソースプログラムをクロスコンパイラで組込用マイコンの実行形式のプログラムを作成する。このプログラムをシリアルインタフェースでマイコンボードに転送・実行してプログラムの動作を確認する。

## 2.1 パソコンの組立

Linuxを市販のパソコンに導入した場合、Linuxが正常に動作しない大きな要因にLinuxの入出力デバイスドライバのプログラム(以後、ドライバと呼ぶ。)の有無が挙げられる。市販のパソコンに合わせて、Linuxのドライバを作成したり、ドライバ組み込んだりするとシステムの設定に時間を要する。

そこで、今回は、配布されているLinuxのシステム

\*1 機械電子部

\*2 材料部

に入っているドライバの適合性を調べ、ドライバに合った入出力のデバイス(カード)等を選択して、パソコンに組み込んだ。製作したパソコンの仕様を表1に示す。

表1 Linux用パソコンの仕様

部品名	性能
CPU	733MHz
メモリ	128M バイト
ハードディスク	20G バイト

## 2.2 Linux の選定

Linux は、オープンソースとして多くのボランティア団体によって開発されているので、幾つかの種類(distribution)がある。本研究では、Linux をマイクロコンピュータのソフトウェアの開発システムとして利用するので、Linux のシステムの細部まで知る必要がある。そこで、比較的プログラミングに関する資料の多い Redhat 社の Redhat Linux を用いた。この Linux を基にマイクロコンピュータの開発システムを構築した。

## 2.3 クロスCコンパイラの生成

研究では、組込用マイコンとして、比較的多く使われている日立製作所の 16bit マイコン H8 を使用した。H8 のクロス C コンパイラは、オープンソースである gcc(GNU Compiler Collection)を用いて以下の手順で生成した。

- (1) gcc の C 言語のソースコードをハードディスクに展開する。
- (2) H8 用のオブジェクトコードが生成されるように設定する。
- (3) Linux の載ったパソコンに標準で入っているインテル社製の 86 系プロセッサ用の gcc を用いてコンパイルし、H8 用のクロス C コンパイラを生成する。
- (4) 生成した H8 用のクロス C コンパイラを実行できるように設定する。

開発システムの構築のため用いたオープンソースは、表2のものを用いた。

表2 用いたオープンソース

オープンソース名	プログラムの種類
gcc	C コンパイラ
binutil	アセンブラ、ローダ

## 2.4 転送プログラムの開発

転送プログラムの開発においては、次の五つの点に考慮した。

- (1) パソコンから組込用マイコンへプログラムを転送して実行できるようにする。
- (2) 転送プログラムの操作が容易に行えるように GUI(Graphical User Interface)を利用して作成する。
- (3) (2)を考慮して Linux のウインドウシステムの一つである GNOME のアプリケーションとして動作するようにする。
- (4) メニューの拡張ができるようにする。
- (5) プログラムの転送は、RS-232C を用いる。

以上のことを満足するように、GTK+<sup>1)</sup>のライブラリを使って転送プログラムを構築した。GTK+は、図2に示す階層構造をとる。GTK+は、MIT で UNIX 用に開発された X ウインドウ用のライブラリである Xlib 上に新たにライブラリを構築して、X ウインドウのプログラムをより開発しやすくしたオープンソースのライブラリである。



図2 GTK+の構造

転送プログラムは、図3に示すフローで構築した。作成した転送プログラムは、大きく分けて三つの処理グループから構成される。

- (1) ウインドウとデバイスの初期化
- (2) メニューの選択
- (3) 各メニューの処理(コールバックルーチンと呼ぶ。)

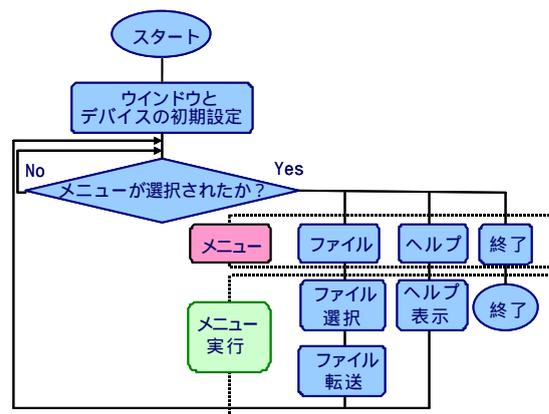


図3 転送プログラムのフロー

ウインドウシステムとデバイスの初期化は、プログラムが実行されると同時に RS-232C とウインドウシステムを初期化し、ウインドウを画面に表示する(写真1)。初期化は、メニューをマウスで選択したときと

RS-232C からデータが入力されたときに、指定した処理ルーチンに飛びイベントの登録を行っている。

ウインドウにある各メニューの動作は、メニューをマウス等で選択することによって、割り込みが発生し、初期化で登録されたメニューの処理ルーチンであるコールバックルーチンが実行される仕組みになっている。転送プログラムのコールバックルーチンの中にメニューの「開く」と「終了」のルーチンが用意されている。「開く」のメニューをマウスで選択するとファイル選択のダイアログ（ファイルセレクションダイアログ）が表示される。ダイアログ上でマウス等を用いてファイル名を選択してから、OK ボタンを押すとダイアログからプログラムに選択されたファイル名が渡され、そのファイルが RS-232C を経由してマイクロコンピュータボード（以後、マイコンボードと呼ぶ。）に転送される。「終了」メニューを選択するとウインドウの終了ルーチンを呼び、ウインドウが閉じて転送プログラムの実行を終える。「ヘルプ」メニューの中にある「バージョン」のメニューは、プログラムのバージョン、作成者名、作成年月日、コメント等を書いたダイアログを表示するようになっている。また、現在の転送プログラムのメニューは少ないが、今後、拡張できるような構造になっている。



写真1 転送プログラムの実行

### 3. 開発結果および考察

#### 3.1 マイクロコンピュータボード

プログラムを実行するために用いたマイコンボードを写真2に示す。このマイコンボードは、市販の組立キットを用いた。これは128KバイトのフラッシュメモリとRAM4Kバイトの16bitの1チップマイコンであるH8により構成されており、外部インターフェースとして、RS-232Cがある。このマイコンボードと生成したクロスCコンパイラを用いて簡単なC言語のプログラムの

実行を行った。



写真2 マイコンボード

#### 3.2 クロスCコンパイル

gcc から生成したクロス C コンパイラを用いて、C 言語のソースコードをコンパイルした結果を図4に示す。用意したC言語ソースプログラム図4(a)は、数行の小さなプログラムである。これを H8 用のクロス C コンパイラを使ってアセンブリ言語のコードを生成した。その結果を図4(b)に示す。結果からC言語のソースコードが H8 のアセンブリ言語になったことが分かる。

```
#include <stdio.h>
int main( )
{
    printf("Hello world\n");
}
```

(a) Cのソースプログラム

```
.section.rodata
.LC0:.string"hello world\n"
.text
.align 4
.globl main
.type main,@function
main:pushl %ebp
    movl %esp, %ebp
    subl $8, %esp
    subl $12, %esp
    pushl $.LC0
    call printf
    addl $16, %esp
    leave
    ret
```

(b) オブジェクトコード(H8)

図4 ソースプログラムと生成されたコード

### 3.3 実行プログラムの生成

組込用のマイコンは、OSを持たないので、C言語のプログラムを実行する場合、図5に示すように、アセンブリ言語プログラムによってC言語の実行環境を整えてからCのプログラムを実行する。このC言語の実行環境を整えるプログラムをCのスタートアップルーチンと呼ばれている。今回これを作成して、アセンブルし、図4のC言語のプログラムをリンクした。そして、オブジェクトコードをマイコンボードに転送できるように16進数の文字コード(ヘキサコード)に変換した。

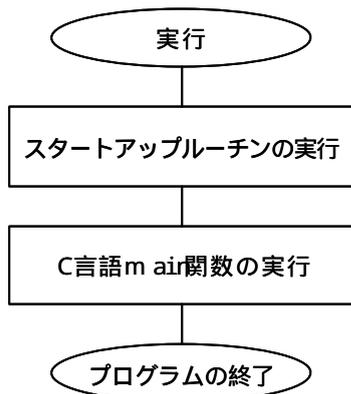


図5 C言語の実行手順

### 3.4 プログラムの転送および実行

プログラムの転送は、写真3に示すように、パソコンとマイコンボードをRS-232Cで接続して行った。



写真3 パソコンとマイコンボード

パソコンで、転送プログラムを実行して、ファイルメニューの「開く」を選択すると写真4に示すファイルセレクションダイアログが表示されて、送りたい実行ファイルを選択するとプログラムの転送が始まる。転送が終了した時点で、マイコン側からプログラムを実行する番地を入力すると写真5のように表示され、プログラムが正常に実行したことが確認できた。



写真4 ファイル選択画面



写真5 プログラムの実行

## 4. 結 び

本研究では、Linuxを用いた安価なマイコンの開発システムを構築した。開発システムは、パソコンと開発用のプログラムからなっている。パソコンは、Linuxを導入するためにパソコン部品を選定して製作した。

今後、この開発システムとLANの付いたマイコンボードを用いて、工場にある機器等の計測データを収集・分析して、ホームページに自動的に公開するシステムを構築していく予定である。

## 参考文献

- 1) Kurt Wall, et al.:Linux Programming Second Edition,P637(2001),Sams Publishing