

1

text : アップウインドテクノロジーインコーポレイテッド 中村憲一
NAKAMURA Ken-ichi nakamura@upwind-technology.com

組み込みLinux業界動向 2004

昨年に引き続き、Linuxは組み込み機器関連の分野でも大変に注目されています。実際にLinuxを搭載した家電製品が多数市場に出回るようになってきました。ここで活躍するのが、組み込みLinuxエンジニアです。本特集では、組み込みLinuxエンジニアを志す皆さんのための情報を紹介していきます。

はじめに

本誌でもよく取り上げられているLinuxシステムですが、最近では、AV機器や携帯電話に搭載されるなど「組み込み」と呼ばれる分野にも進出し、話題になっています。

本特集では、一般に「組み込みLinux」と呼ばれる組み込みに特化したLinuxシステム全般について、開発するにあたって必要となる知識、開発手法、開発手順などを中心に紹介します。

まず、本章では、「組み込みとは何なの?」「組み込みLinuxはどこでどのように使われるの?」「組み込みLinuxにもディストリビューションが存在するの? どうやってインストールするの?」「組み込み業界ってどんな業界なの?」など日頃よく聞かれることの多い疑問を解決していきたいと思えます。

組み込みとは

「組み込み」とは何でしょうか? 英語ではEmbeddedと言いますが、一般の方にはEmbeddedだけでは意味が通じないことが多いのでEmbedded System, Embedded Softwareなどと言う場合が多いです。Embeddedを辞書で調べるとembedの過去/過

去分詞形で「はめ込まれた、組み込まれた、埋め込まれた」という意味であることがわかります(邦訳文書の中には、「埋め込みシステム」「埋め込みソフトウェア」などのように訳されている文書もたまに見受けられます)。さらに調べると、embedは「~の中に入れる」という意味の動詞をつくる接頭語enの異形であるemとbedにより構成されていることがわかります。ここで、bedには寝台という意味がありますが、土台、路盤という意味もあります。

つまり、embeddedは「基盤の中に入れられた」という意味になります。ここでいう基盤とは、もちろんプリント基盤のことを指します。よって、「組み込み」とは、厳密に言うとプリント基盤に実装されるプロセッサ/ROM/RAM/FPGA/CPLD/抵抗/コンデンサ/ディップスイッチ/LED/各種コネクタなどのハードウェア、さらにソフトウェアを総称したものと定義することができます。この定義では、読者のみなさんが普段利用しているPCも「組み込み」機器になります。

しかし、これでは何もかもが「組み込み」機器になってしまいますので、従来は携帯電話/PDA/AV機器などのような、キーボード/マウス/ディスプレイ/ハードディスクを持たない電子機器を総称して「組み込み」機器と呼んでいました。

1

2

3

4

5

Appendix

ところが、最近では、HDDビデオレコーダ/HDDカーナビゲーションシステム/キーボードを備えたPDA/GUIをポインタでマウスのように操作する携帯電話など、これらのデバイスを備える電子機器が当たり前になってきましたので、「組み込み」を定義することがますます難しい問題になっています。

組み込み機器とOS

組み込み機器にOSは必要か

そもそも「組み込み」機器にOSは必要なのでしょうか？ また、「組み込み」機器用のOSにはどのようなOSがあるのでしょうか？

社団法人トロン協会 (<http://www.assoc.tron.org/jpn/>) と社団法人日本システムハウス協会 (<http://www.jasa.or.jp/>) は、日本でどのようなOSが使用されているかを調査しており、その成果をトロン協会のWebページにおいて「組み込みシステムにおけるリアルタイムOSの利用動向に関するアンケート調査報告書」(<http://www.assoc.tron.org/jpn/research/index.html>) として公開しています。

この報告書を見ると、従来は主流だった4ビット/8ビットのCPUは現在ではほとんど使われておらず、16ビット/32ビットのCPUが主流になっています。32ビットのCPUのうち、具体的には、ARM/MIPS/PowerPCが増えています。Z80/68000/H8/Sが主流だった時代は、すべてのプログラムをアセンブラで書くのが当たり前でしたが、このように32ビットのCPUが主流になった現在では、チューニングなど特別な場合を除き、アセンブラでプログラムを書くことはほとんどなくなったと言っても過言ではありません。そのうえ、プログラムサイズも4Mバイト未満のものは少なくなり、4Mバイト以上のものが増えてきているため、とてもすべてをアセンブラで書くわけにはいかない状況になっています。

そこで、CやC++言語でプログラムを書くことに



なるのですが、C言語はもともとUNIXシステムを構築するために開発された言語であり、「組み込み」システムを構築するために開発された言語ではありません。そのため、CやC++言語のみでゼロからスケジューリングや割り込み処理などを記述するのは非常に手間のかかる作業になってしまいます。そこで、「組み込み」機器用のOSの助けを借りるというアイデアがでてきたのです。同調査の2002年度と同調査報告書(2003年11月12日発行)では、「組み込み」機器用のOSではITRON仕様OSが38.6%、Linuxが10.9%、VxWorksが7.2%、Windowsが3.4%、Windows CEとOS-9(Mac OS 9とは異なるOSであることに注意してください!)がそれぞれ2.8%を占めると報告しています。

そのほかのOS、とくにリアルタイムOSについては、サンリツオートメーション(株)が運営している「リアルタイムOS情報」のWebページ(<http://www.sanritz.co.jp/realtime/>, 図1)にまとめられています。

TRON vs. Linux ?

最近、新聞/週刊誌/テレビなどの報道を見ると坂村健 vs. Linus Torvaldsといった見出しで、TRONとLinuxがそれぞれのシェアをめぐる戦いを繰り広げているように取り上げられていますが、実際はそうではありません。記事としては確におもしろい筋書きとは思いますが、そもそもTRONとLinuxを比べること自体が間違っているのです。

TRONの1つの種類である^{アイトロン}は、ハードリアルタイムと呼ばれる処理に対応できるのに対し、

第 2 特集 組み込みLinuxエンジニアになるための基本学習帳



Linuxではソフトリアルタイムと呼ばれる処理にしか対応できませんので、そもそも用途が異なります。

たとえば、ITRONがロボットなどの制御をマイクロ秒単位でできるのに対し、Linuxではミリ秒単位がやっとです。しかもそのミリ秒さえ、必ず実行できるという保証はありません。言い換えると、このような用途にはLinuxはそのままの状態では使い物にはならないのです。

TRON から考える

では、ここでTRONについて少し整理しておきましょう。TRONには、BTRON / CTRON / ITRON / マイクロアイトロン μ ITRONという種類があります。そして、その中でも μ ITRONが最も有名です。

μ ITRONは、デファクトスタンダードのリアルタイムオペレーティングシステムとして組み込み業界の人々の間では常識でしたが、2003年、NHKのTV番組『プロジェクトX』シリーズで取り上げられて以来、一般の人々の間でも一躍有名になりました。

μ ITRON仕様は非常に優れたAPIの仕様であり、各メーカーがこぞって採用してきましたが、バブル経済の崩壊と共に開発期間の短縮化、開発コストの削減という大きな課題に直面したため、次のような問題が表面化してきました。

- OSの実装方法が未定義である
- デバイスドライバの標準化がされていない
- 技術者が不足している

- 教育機関がない

これらの問題は次のように言い換えることができます。

- アプリケーションの再利用ができない
- デバイスドライバの再利用ができない
- 開発する人がいない
- 開発者を育成したくてもできない

つまり、 μ ITRON仕様が良い仕様であることは事実ですが、アプリケーションやデバイスドライバの融通が利かず、開発者も少ないという状況に陥っています。これは、機能を追及するためにAPIの標準化に重きを置き過ぎた代償とも言えるでしょう。

現在ではこれらの問題を解決するためにTOPPERSプロジェクト (<http://www.toppers.jp/>) やT-Engineプロジェクト (<http://www.t-engine.org/>, 図2) が活発な活動を行っています。

2004年以降これらのプロジェクトの開発成果が続々と発表される予定ですので、今はこれらのプロジェクトに大きく期待したいと思います。

ここでは誌面の都合上、Windows CE .NETについては触れませんが、Windows CE .NETもTRONやLinuxに匹敵する実力を備えています。今後はこれら3つのOSが「組み込み」機器のOSとしてシェアを分かち合っていくことになるでしょう。

組み込みLinux とは

さて、「組み込み」や組み込み用のOSについて少し理解できたところで、「組み込みLinux」について説明したいと思います。「組み込みLinux」とは、その名のとおり前節で説明した「組み込み」に特化したLinuxのことです。

なぜLinux?

サーバ分野でのLinuxの採用が拡大するにつれ、各メーカーでもさまざまな問題を解決するためにLinuxを「組み込み」機器に採用することが検討されはじめました。

1

2

3

4

5

Appendix

表1 組み込みLinuxのディストリビュータ

製品名	開発元	URL	日本での販売元
MontaVista Linux	MontaVista Software, Inc.	http://www.mvista.com/	モンタビスタソフトウェアジャパン(株)
BlueCat Linux	LynuxWorks, Inc.	http://www.lynuxworks.com/	日進ソフトウェア(株)
TimeSys Linux	TimeSys Corp.	http://www.timesys.com/	(株)日新システムズ
RTLinux	FSMLabs, Inc.	http://www.fsmlabs.com/	(株)エフエスエムラボ・ジャパン
OpenPDA	Metrowerks Corporation	http://www.metrowerks.com/	メトロワークス(株)
Silicon Linux	(有)りぬくす工房	http://www.si-linux.org/	左に同じ
axLinux, zxLinux	(株)アックス	http://www.axe-inc.co.jp/	左に同じ
uLinux	リネオソリューションズ(株)	http://www.lineo.co.jp/	左に同じ

表2 組み込みLinuxのシステムインテグレータ

会社名	URL	備考
(株)イーエルティ	http://www.emblit.co.jp/	2001年5月設立
米 Upwind Technology, Inc.	http://www.upwind-technology.com/	2002年8月設立

その結果、組み込みLinuxの世界でも数え切れないほどのディストリビューションが世界中で発表され、商用目的のディストリビュータ(表1)やシステムインテグレータ(表2)も新しく設立されました(Linuxと言えばサーバ分野ではとても有名なRed Hat, Inc.ですが、実は2002年の半ばまでは組み込みLinux用のパッケージも販売していました。しかし、現在は販売を停止しているようです)。

なぜ、これほどまでにLinuxが人気なのかというと、Linuxの採用には以下のようなメリットが考えられるからです。

- オープンソース
- ロイヤリティフリー
- アプリケーション開発者の確保が容易
- アプリケーションやデバイスドライバが簡単に再利用できる

これを言い換えると次のようになるでしょう。

- 自社製や他社製のITRON仕様OSを使用していたときと同じようにカーネル内の処理を把握することができる
- カーネルやライブラリにロイヤリティが発生しないので製品価格を下げるができる
- UNIXプログラミングの経験者であれば上位のアプリケーションが書ける
- ほかのプロジェクトで開発したプログラムを再利用できる

- デバイスの供給元にデバイスドライバを要求できる

このように考えると、組み込み機器開発にとって、Linuxはまさに夢のようなOSなのです。

組み込みLinuxの採用例

すでに製品に組み込まれている例(表3)が複数ありますので、読者の皆さんの中には、すでに使用している方も多いでしょう。ここで挙げた中には、RTLinuxやTimeSys Linuxのようにハードリアルタイム処理にも対応できるよう改善されたディストリビューションもあり、TimeSys LinuxはNASAの火星探査機「マーズパスファインダ」に採用されるなどの実績をあげています。

しかし、Linuxにはメリットばかりではなく、反対にGPLにみられるような厳密なライセンス体系に対応する必要があるなど、手間のかかる部分もあります。これらの問題については本特集5章で詳細に説明しますが、実際にLinuxを製品に採用しているソニーでは、Source Code Distribution ServiceのWebページ(<http://www.sony.net/Products/Linux/>、図3)において、製品に採用された組み込みLinuxのソースコードを公開する、という方法でライセンス規約を遵守しています。

第 2 特集 組み込みLinuxエンジニアになるための基本学習帳

表 3 組み込み Linux が採用されている製品の例

メーカー	製品カテゴリ	製品名 / 型番
ソニー(株)	ホームシアターシステム	CoCoon (NAV-E900/E600)
	ビデオレコーダ	CoCoon (CSV-S55/CSV-E77/CSV-P500/NDR-XR1)
	デジタルテレビ	WEGA (KDE-P50HZ1/KDE-P42HZ1)
	家庭用ゲーム機	PlayStation BB Navigator
	ブロードバンドAVルータ	HN-RT1
	ビデオスイッチ	CCP-8000/CCP-9000
シャープ(株)	パーソナルサーバ	Galileo
	PDA	ザウルス (SL-A300, SL-B500, SL-C700/750/760)
松下電器産業(株)	ビデオレコーダー	Broadnow (FZ-BB1000)
パナソニックコミュニケーションズ(株)	ホームゲートウェイ	ホーム・IPセキュリティ・ゲートウェイ
	ネットワークゲートウェイ	デジタルネットワークゲートウェイ (DN-G200)
(株)NTT-ME	ホームサーバ	LivingGate i
日本電算機(株)	家庭用サーバ	iBox ブロードメディアサーバ
日本電気(株)	ホームAVサーバ	AX-10
プラット・コミュニケーションコンポネンツ(株)	ブロードバンド・ターミナルボックス	BTBox
ぶらっとホーム(株)	マイクロサーバ	OpenBlockSシリーズ

図 3 Source Code Distribution Service (ソニー)



情報源

最後に、組み込みLinux関連の情報が得られる情報源を紹介しておきます。

Web ページ

- 社団法人日本システムハウス協会 (<http://www.jasa.or.jp/>, 図4)
2002年3月に「Embedded Linuxにおける技術動向」と題した調査報告書を作成しており、本報告書の抜粋を同Webページよりダウンロードできます。
- Embedded Linux Consortium (<http://www.embedded-linux.org/>)

組み込み機器に要求される機能をLinuxカーネルで実現するための仕様を、APIレベルで決定し、Linuxを広く普及させることを目的としている団体です。2003年12月8日現在、同WebページからEmbedded Linux Consortium Platform Specification (ELCPS) v1.0をダウンロードできます。

- Linux Devices.com (<http://www.linuxdevices.com/>)

Linuxが製品や試作品に採用された世界中の例を紹介しているホームページです。

- 日本エンベッドリナックスコンソーシアム (<http://www.emblix.org/>, 図5)

Linuxの組み込み業界への普及 / 浸透 / 組み込みLinux周辺技術の標準化などを目的として活動しているNPO法人です。

- CE Linux Forum (<http://www.celinux.org/>, 図6)

次世代デジタル家電向けのLinuxとして注目されているCE Linuxを開発している任意団体です。2003年12月8日現在、031030版が公開されています。IBMやソニーなども参加しています。

1

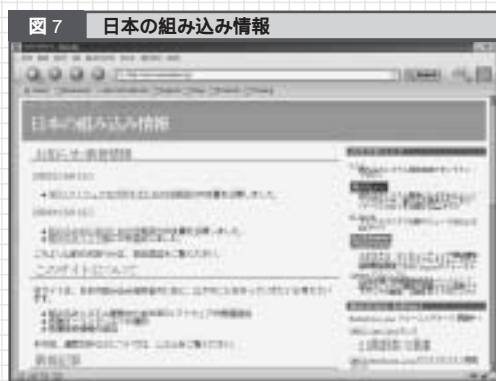
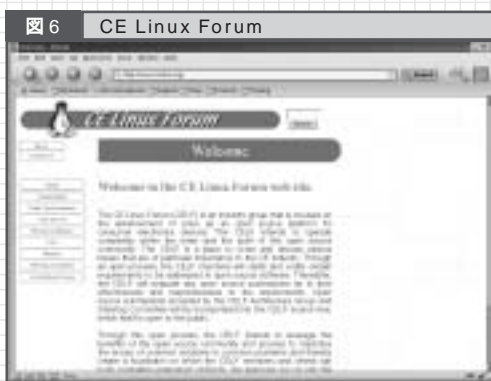
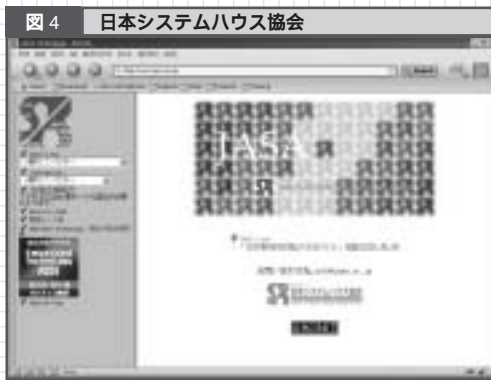
2

3

4

5

Appendix



- 「日本の組み込み情報」(<http://www.embedded.jp/>, 図7)

アップウィンドテクノロジー・インコーポレイテッドが提供する「組み込み」や「組み込みLinux」に関するWebページです。各種技術情報やメーリングリストが用意されており、組み込みシステム開発ツールであるGNUWingがダウンロードできます。

書籍

- 『詳解 Linux カーネル 第2版』
(Daniel P. Bovet, Marco Cesati 著 / 高橋 浩和監訳 / 杉田 由美子, 高杉 昌督, 畑崎 恵介, 平松 雅巳, 安井 隆宏訳 / オライリー・ジャパン / ISBN : 4-87311-133-1)
- 『Linux デバイスドライバ第2版』
(Alessandro Rubine, Jonathan Corbet 著 / 山崎康宏, 山崎邦子, 長原宏治, 長原陽子訳 / オライリー・

ジャパン / ISBN : 4-87311-081-5)

- 『組み込みLinuxシステム構築』
(Karim Yaghmour 著, 林 秀幸訳 / オライリー・ジャパン / ISBN : 4-87311-161-7)

最後に～本特集の構成

次章以降では、2章で実際の組み込みLinuxエンジニアへのスキルアップ術、3章で実際の開発ツール(GDBなど)の利用方法、4章で実践的な開発工程、5章でLinuxエンジニアが知っておくべきライセンスと法について紹介します。また、Appendixとして、開発ツールや各社のトレーニングプログラムについて紹介しますので、参考になさってください。SD