

# SuperHにおけるGNU/Linuxディストリビューションの開発

Development of GNU/Linux distributions on SuperH

小島 一元<sup>1)</sup> 杉岡 利信<sup>2)</sup> 新部 裕<sup>3)</sup>  
野澤 寿晴<sup>4)</sup> 吉井 卓<sup>5)</sup> 八重樫 剛史<sup>6)</sup>

Kazumoto KOJIMA Toshinobu SUGIOKA Yutaka NIIBE  
Toshiharu NOZAWA Takashi YOSHII Takeshi YAEGASHI

- 1) (〒244-0002 横浜市戸塚区矢部町 946-31 E-mail: kkojima@rr.iij4u.or.jp )
- 2) (株)アイ・ティー・オー(〒573-0163 枚方市長尾元町 7-4-8 E-mail: sugioka@itonet.co.jp)
- 3) 産業技術総合研究所情報処理研究部門(〒305-8568 茨城県つくば市梅園 1-1-1 中央第二 E-mail: gniibe@m17n.org)
- 4) (株)日立超 LSI システムズ プラットフォーム設計部(〒187-8522 東京都小平市上水本町 5-22-1 E-mail: nozawat@hitachi-ul.co.jp)
- 5) (株)日立超 LSI システムズ プラットフォーム設計部(〒187-8522 東京都小平市上水本町 5-22-1 E-mail: yoshii-takashi@hitachi-ul.co.jp)
- 6) (株)ナムコ 開発技術部(〒221-0031 神奈川県横浜市神奈川区新浦島町 1-1-32 ニューステージ横浜 16F E-mail: yaegashi@rd.namco.co.jp)

**ABSTRACT.** DODES is the network system for cross development of GNU/Linux distribution. Because of its features, source code availability and the world wide cooperation, GNU/Linux is getting used in many environments, scale up to the mainframe and down to embedded system. For the development of embedded targets, the performance of target system (which is relatively slow) could be the bottle-neck of development and major problem. With DODES system, cross compiling environment on high performance server is used to generate the binary and the binary runs transparently on target connected with network, which is quite convenient and resolve the bottle-neck.

とが求められる。

## 1. 背景

これまで GNU/Linux システムは PC のプラットフォームで稼働してきており、この分野では技術的側面、社会的側面の双方において高い実績と定着した評価が既にある。そして、新たな技術的進展として、組み込み機器の分野でも GNU/Linux 用いることが可能となっており、今後、家電製品などの分野にも応用が広がることが期待されている。

組み込み機器の分野に GNU/Linux システムを利用するためには、研究レベルの現在のシステムを実際の製品に利用する段階のソフトウェアとしてまとめていく必要がある。具体的には、安定して稼働することが確認されたソフトウェアをパッケージとして配布し、稼働実績を広めていくことが必要となる。特に、このパッケージ作成に際して、テストスイートを一通り流して、動作の確認を行なう技術が重要なポイントとなる。

ディストリビューションは、異なる組織、異なる環境で作られたソフトウェアを一つの環境としてまとめあげるものであり、システム全体として整合性を持ち、一貫性を持った構成を取ることが重要である。そして、その構築環境としては、依存関係が整理され、再現性の高い構成で、効率よくパッケージの構築とテストができるこ

## 2. 目的

DODES は複数のターゲットマシンを、ホストマシンからネットワークを用いて利用するクロス開発環境のネットワークシステムを構築し、テストスイートをターゲットマシンに分散させてテストを実施する技術を実装するソフトウェアとして開発された。これはまたクロス環境におけるテストシステムの構築を行ない、膨大なテストスイートを多くのターゲットマシンにばらまいて効率良くテストを実施する技術を実装し、ディストリビューション作成など大規模な作業の効率を向上させることを目的としている。

## 3. DODES システム

### (1) 概要

DODES では、ホスト計算機からの遠隔実行要求を受け、ターゲット計算機上でプログラムが稼働する。この際、複数のプログラムの全体の実行時間を短縮させるために、複数あるターゲット計算機において、負荷分散を行う。この負荷分散はプロセスの実行を単位とする荒い粒度の負荷分散であり、プロセスのマイグレーションは行わず、負荷の分散はプロセスの投入時にその時点で

の各ターゲット計算機の負荷の情報に基づいて行なわれる。

DODES の主な利用目的は、ディストリビューションの作成とそのテストにあるため、投入されるプロセスの数はターゲット計算機の数と比較して多数であり、プロセス間の関連が(特別の場合を除いては)少ない、独立に稼働するプロセス群であるという性質がある。このため、プロセスの投入時にターゲット計算機を選択する、荒い粒度の負荷分散のサポートで、実用上十分と考えられる。

## (2) 実装

DODES ではホスト計算機上であるプログラムを実行する場合にそれがターゲット計算機のためのものであることを検知し、複数あるターゲット計算機のどれかの上での遠隔実行を行う。ターゲット計算機とホスト計算機は NFS によってファイルシステムを共有しておりターゲット計算機におけるファイルのパス名がホスト計算機での対応するファイルのパス名が同一または簡単なマッピングで対応するように構成される。

UNIX 系オペレーティングシステムでの実行機能は `execve` と呼ばれるシステムコールに集約されるため、カーネルまたは共有ライブラリレベルでこの `execve` の中に遠隔実行機能を実現することで既存のプログラムを再コンパイルすることなく遠隔実行機能を付加することも可能になる。

現実装ではシステムライブラリである GNU libc の `execve` システムコールのラッパー関数において実行ファイルがターゲット計算機用のものであるかどうかを判定し負荷分散機能の下で選択されたターゲット計算機上で遠隔実行を行う。この場合に GNU libc でも実装されている共有ライブラリのプリロード機能が非常に有効であった。

## (3) 実行例

DODES を稼働させた例として GNU Compiler Collection (GCC)における C コンパイラのリグレッションテストスクリプトの実行に要した時間を上げる。このテストスクリプトではターゲット用コンパイラにより多数の小さなソースプログラムのコンパイルを行いその約半数についてはコンパイルされた実行可能プログラムのターゲットでの実行を伴う。

GCC のバージョンは 2.97 でテストの総数 13154 個に対しテストの所要時間はホスト celeron 600MHz 主メモリ 256M byte とターゲット SH-4 200MHz 主メモリ 64M byte からなる最も単純なシステムで 28 分 20 秒であったのに対しターゲット計算機のみでのテストには 185 分 13 秒を必要とした。このテストスクリプトの実行時間の多くはコンパイル時間に費されるが、ターゲットが並列化されない DODES でもこのような場合には非常に有利であるといえる。なお GCC のバージョンは 3.x では C コンパイラのテストの総数が 16000 弱に達しテストの所要時間も 2.97 の 2 倍程度が必要になっている。

このように開発環境として最も重要である GCC のテストが DODES 環境で効率良く可能となった意義は大きいと思われる。

他の実行例としては GNU libc, emacs, Perl, Ruby などのクロス環境での構築がある。これらはターゲットでの実行プログラムを実際に走らせて構築を行うため、通常のクロス環境では完全な構築は困難であったが、DODES ではターゲット上での構築とほとんど変わらない操作で、高速な構築ができる。

## (4) 結論と課題

DODES のアイデア自体は非常にシンプルでその構成要素自体はよく知られたものであるが、クロス開発環境で一般的ないくつかの場合に非常に強力な手段となることがわかった。この手法の有効性はターゲットとホストでほぼ同一の GNU/Linux という環境が用意できるという組み込み GNU/Linux システムではごくあたりまえの特性からの帰結だと考えられる。

DODES プログラムの中間バージョンのソース及び簡単なドキュメントは

<http://dodo.nurs.or.jp/~kkojima/dodes/>  
に置かれている。CVS による最新のリソースの提供を [m17n.org](http://m17n.org) で行うように移行作業を行っている。

現行の DODES における課題として複数アーキテクチャのターゲットの同時使用を許すことや遠隔実行の安全性の問題をあげることができる。これらの課題を解決したシステムの開発を予定している。

## 4. まとめ

自由にソースコードの形態で流通できるソフトウェアが豊富になり、それら外来のソフトウェアを集めて構成することで高性能のシステムを短期間に実現することが、比較的簡単に可能となってきた。その構成の元となる集大成がディストリビューションであり、ネットワークサービスを実装するサーバーや研究開発システムを中心に、広く用いられてきた。

こうしたディストリビューションの利用は、組み込みシステムにも適用可能であり、高機能で複雑なシステムを構築する際に有効な手段であることは間違いない。組み込みシステムにおいては、ターゲットシステムが(比較的)性能が低いことがディストリビューション構築時のボトルネックとなるが、DODES システムの工夫により、高い生産性をあげることができる。

## 5. 参加企業及び機関

なし。

## 6. 参考文献

- [1] 新部 裕ほか, Linux on Hitachi SuperH(TM) Processor, <http://linuxsh.sourceforge.net/>.
- [2] 新部 裕, GNU/Linux on SuperH プロジェクト最近の話題, インターフェイス, 2000 年 4 月号 P.201-208, CQ 出版