

Embedded NetBSD を生成するプログラムの展望 Prospects of a program to generate Embedded NetBSD

榎本 優樹[†] 深町 賢一[‡]
Yuuki Enomoto Ken'chi Fukamachi

1. はじめに

NetBSD や Solaris に代表される Unix 系オペレーティングシステム (以下 Unix) は、カーネルの構造やユーザランド側の機能など学習する内容が多岐にわたるソフトウェアである。本研究では Unix のユーザランド側に注目し、クロスビルド可能な Unix である NetBSD のソースセットから、USB メモリや SD カードといった取り外し可能なメディアにオペレーティングシステムを書き込む配布形態である Live Image を作成するプログラム「fdgw2」の開発を目的とした。fdgw2 によってユーザは任意の Unix コマンドやディレクトリ構造を持つ NetBSD Live Image を作成でき、カーネルを除く Unix の構造を俯瞰する効果が期待される。また fdgw2 はすべての Unix 環境で実行でき、ユーザが NetBSD をインストールする必要がない。

本稿では fdgw2 の概要を述べ、今後の展望を考察する。また fdgw2 では本来組み込み開発やインストーラの構築をするために使われた手法を用いたことから、このプログラムによって生成される Live Image を「Embedded NetBSD」と呼ぶ。

2. 開発背景

Unix の操作方法を学習するには、ノート PC や仮想マシンに Unix をインストールし手を動かしながら動作を確認することが望ましい。同様にカーネルや Unix コマンド、ファイルシステム、ファイル、ブートストラッププログラムがどのように構成されて 1 つのシステムとなっているかを知るには、Unix のソースセットから Live Image を作成することが望ましいと考えた。また Live Image を単純な構造にすることがユーザの理解を容易にすることに繋がると考え、組み込みシステムやインストーラの作成に用いられる手法の採用を決定した。

組み込みシステムとして小さな NetBSD Live Image を構築する手法は既に確立されている。The NetBSD Project の Web ページ[1]では NetBSD を用いた組み込み開発について書かれた Web ページへのリンクが掲載されている。しかしいずれも作業を自動化するに至っていなかった。

make コマンドを使用し、上記の Web ページに掲載されているような手法で Live Image を作成するプログラムとして深町による「fdgw」がある[2]ものの、NetBSD-2.0 以降開発が停滞しており NetBSD がインストールされたマシンでなければ実行できないものであった。そこで我々は fdgw をもとに NetBSD-7.0 のソースセットを用い、ビルド環境を NetBSD に限定しないプログラムを開発し「fdgw2」と名付けた。

3. fdgw2

3.1 概要

fdgw2 は複数の Makefile から構成されるプログラムである。C や Perl といった高水準言語は用いておらず、NetBSD のソースセットに付属しているクロスビルド用スクリプト build.sh から生成されるツールチェーンに含まれる Unix コマンドを使用している。build.sh のツールチェーンを用いることによりオペレーティングシステムやアーキテクチャに依存せず fdgw2 を実行できるという利点がある。我々は Debian/GNU Linux-8.2.0 amd64 上にて fdgw2 を実行し、NetBSD-7.0 i386 カーネルが動作する Embedded NetBSD が作成でき、USB メモリから起動することを確認した。

現在 fdgw2 はフリーソフトウェアとして GitHub 上で公開している (<https://github.com/user340/fdgw2.git>)。

3.2 ビルドの手順

fdgw2 を実行した際、以下の手順で Embedded NetBSD が生成される。

1. crunchgen コマンドで複数のプログラムを単一のプログラムにまとめる。この際、設定ファイルを編集することにより任意の Unix コマンドを追加・削除できる。
2. fdgw2 独自の config を元にカーネルをビルドする。
3. makefs コマンドでルートファイルシステムを作成し、mdsetimage コマンドでルートファイルシステムをカーネルに埋め込む。
4. 再度 makefs コマンドで指定されたディレクトリツリーを元にファイルシステムイメージを作成し、installboot コマンドでブートストラップを書き込む。

作成された Embedded NetBSD は USB メモリに書き込み、USB メモリからシステムをブートすることで利用できる。標準では NetBSD の/rescue を参考に、fsck や disklabel といったシステム復旧用のコマンドがインストールされている。Embedded NetBSD の構造を図 1 に示した。

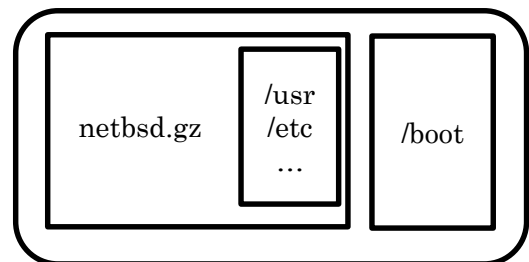


図 1 Embedded NetBSD の構造

[†] 千歳科学技術大学大学院光科学研究科

[‡] 千歳科学技術大学

4. 考察

4.1 fdgw2 の課題

fdgw2 は build.sh によりクロスビルド可能なプログラムであるが独自の config を使用している都合上、カーネルは i386 アーキテクチャ以外をビルドできない。安価かつ高性能なハードウェアが多く存在する今日においてカーネルのサイズを、柔軟性を犠牲にして縮小していくことの利点は少ない。したがってインストールイメージのカーネルが用いる config を使い、任意のアーキテクチャに対応できるように修正すべきである。

また Linux ディストリビューション上で fdgw2 を実行する場合、コンパイラや BSD 版 make などのパッケージが必要であるが、ディストリビューションごとにパッケージ管理システムが異なり、パッケージの名称が統一されていない問題がある。対応に限界はあるものの、fdgw2 実行に必要なパッケージを自動でインストールする、configure のようなプログラムを検討しなければならない。

4.2 Unix の知識調査

fdgw2 は Unix の構造を学習する補助教材として活用が見込まれると考えている。したがってその評価は Unix の初学者を対象とした何らかの講習をおこない、講習前後の習熟の度合いを比較することでなされるべきである。初めに我々は、図 2 に示す設問を回答させる調査を本学のオペレーティングシステムの講義を履修している学部 2 年生、3 年生を対象におこなった。全 12 の設問のうち 1 から 4 は

1. Unix がブートする際、ブートストラップ処理の後に(1)が呼び出される。
2. カーネルのシステムメッセージバッファを出力するのは(2)コマンドである。
3. ディスクのジオメトリ情報を編集し、ディスクにブートストラップをインストールするには(3)コマンドを用いる。
4. 複数のプログラムを 1 つのバイナリにまとめるためには、BSD では(4)、Linux ディストリビューションでは(5)を用いる。
5. Makefile に記述されたコマンド列を実行するには(6)コマンドを用いる。
6. イメージバイナリを USB メモリに書き込むには(7)コマンドを用いる。
7. コマンドの出力を別のコマンドの入力として渡したいとき、(8)と呼ばれる機構を用いる。
8. コマンドの出力をファイルに書き込みたいとき、(9)と呼ばれる機構を用いる。
9. Unix コマンドのマニュアルを読むには(10)コマンドを用いる。
10. ディレクトリ間を移動するには(11)コマンドを用いる。
11. C 言語プログラムをコンパイルするには(12)コマンドを用いる。
12. Unix の標準的なコマンドインタプリタとして(13)コマンドがある。

図 2 Unix の総合知識に関する調査の設問

fdgw2 で使われるコマンドを問う上級、5 から 8 は日常的に Unix を利用していれば理解できる中級、9 から 12 は学部 1 年生で習得する内容である初級の問題として設定した。

結果、履修生 18 名のうち 9 名から回答を得ることができた。正答者が多かった設問順に 10 の 6 名、11 の 5 名、9 の 2 名、2 と 5 の 1 名であり、その他すべての設問は正答者がいなかった。この結果から、Unix 端末上でおこなうプログラミング演習で習得した範囲のことは理解できているものの、シェルインタプリタの基本的な機能や中級以上の事柄については把握していない傾向にあることが分かった。

4.3 今後の展望

本学の学部生を対象とした調査を踏まえ、今後は fdgw2 を用いた Unix 教材の作成を進める。Unix の学習支援を目的としたソフトウェアは UNIX コマンド学習支援システム [3] やシェルコマンド学習支援ゲーム [4] が挙げられ、藤原らは評価実験によって肯定的な評価が得られたと報告している。fdgw2 は NetBSD-7.0 への対応や機能の拡張に多く労力が割かれたためユーザビリティが考慮されていない。教材として特に視覚的な満足度を高めることにより肯定的な評価が期待されるだろう。

5. 結論

Unix 全体の構造を俯瞰できる効果をねらい、NetBSD のソースセットから Embedded NetBSD を作成するプログラム fdgw2 を開発した。また本学で OS の講義を履修している学部生を対象として Unix の総合知識に関する調査をおこない、Unix の習熟度の度合いを得ることができた。今後はこの調査結果を元に、視覚的な満足度を高めることで fdgw2 の教材としての有効性を評価する予定である。

謝辞

本研究にご指導頂いた深町 賢一講師、小松川 浩教授、また NetBSD の開発グループである The NetBSD Project 様と日本の NetBSD ユーザ会である Japan NetBSD User's Group 様に厚く感謝申し上げます。

参考文献

- [1] <http://www.netbsd.org/about/embed.html> (2016/06/20).
- [2] 深町賢一, “ワンフロッピールータを作ろう!”, NETWORK WORLD (2003).
- [3] 藤原 啓, 長瀧 寛之, 大下福仁, 角川 祐次, 増澤 利光, “相互学習に対応した UNIX コマンド学習支援システムの構築”, 情報処理学会第 73 回全国大会 (2011).
- [4] 佐藤 将真, 西田誠幸, “シェルコマンド学習支援ゲームの開発”, 情報処理学会第 77 回全国大会 (2015).