

自由ソフトウェア活動 と組込機器応用

新部 裕
特定非営利活動法人フリーソフトウェアイニシアティブ
/ 産総研



2009-03-05 組込技術とネットワークに関するワークショップ ETNET2009

目次

- 自己紹介
- 自由ソフトウェア
- 組込機器応用概観
- GNU 開発環境
- GNU/Linux 組込応用
- 僕らのチャレンジ
- FSIJ USB Token
 - GNU Privacy Guard
 - PCB 設計
 - AVR-USB
 - CCID Protocol
 - OpenPGP Protocol
 - RSA or ECC 暗号

自己紹介 (1)

- (独)産総研 (2000-)
- IPA (2001-2004)
- 自由ソフトウェアの研究開発と利用の推進
- FSIJ (2002-)
 - Linux-M32R.ORG
 - CODEblog.ORG
 - Google SoC
 - U-20 プログラミングコンテスト
 - GPLv3 国際委員会



GNU Project 開発者 (1989-)
Linux Kernel 開発者 (1993-)
Debian Project 開発者 (2005-)

自己紹介 (2)

(独)産総研 自由ソフトウェア武門研究グループ長
特定非営利活動法人フリーソフトウェアイニシアティブ理事長

- 1989 GNU Emacs 開発: Mule, mlh, eggv4
- 1994 ICOT Free Software
- 1999 GNU/Linux on SuperH 計画 創始者
- 2001-2003 未踏ソフトウェア創造事業 PM
- 2003-2005 オープンソース事業 IPA 参与
- 2005-2007 CODEblog 研究開発
- 2008 FSIJ USB Token 研究開発

GNU/Linux on SuperH の成果を利用した製品群

- ルータ, データアキュイジョン, NAS, ...



• 10年前は、これが challenge でした

いまや Consumer Electronics 向け GNU/Linux 利用いろいろ

- PDA
- 携帯電話
- 液晶 TV
- ハードディスクビデオレコーダ
- ルータ
- Netbook
- ...

自由ソフトウェアって何？ 利用者の自由と社会の連帯



- 0: 制限なくプログラムを実行できる
- 1: 研究し、必要に応じて改造できる (*)
- 2: 複製を配布でき、隣人を助けられる
- 3: 改善し、その改善をコミュニティ全体の利益のために公開できる (*)

(*) ソースコードが利用できることが前提

特徴： 国際協調開発、開放型の開発

自由ソフトウェアの歴史

- 1983: GNU プロジェクト始まる
- 1985: FSF 設立 開発環境として広まる
- 1991: Linux の開発始まる
- 1998: オープンソース運動勃発
- 199X-: たくさんのソフトウェアが自由に！
Apache, Firefox, MySQL, 00o, ...
- 200x: 各国で政府の関心を引くようになる
組込機器応用が広がる (実行環境)
- 僕の活動では ...
- 2003: 1st アジア OSS シンポジウム
- 2008: 6th CodeFest アジア

僕の活動 (1) 自由なコンピューティングの追求

- GNU
 - ブートローダ: GRUB
 - 開発環境: GDB, GCC, Binutils...
 - ライブラリ: GNU C ライブラリ
 - ライセンス: GPLv3
- Linux
 - ネットワークドライバ / プロトコルスタック
 - 組込応用向け CPU サポート: SuperH, M32R
- Debian - 主に組込応用
 - 最近は KUROBOX/PRO

僕の活動 (2) 自由ソフトウェアと社会

- 研究開発の成果を自由ソフトウェアとして公開できる道を探る
 - 開放型ソフトウェア
 - RING プロジェクト
 - 未踏ソフトウェア ... ここで成功
 - (独)産総研で
- 若い人に自由ソフトウェア活動を伝える
 - CodeFest アジア, Google SoC
- 闘い: この自由を守る FSIJ GPLv3

産業応用との衝突

- 既存のビジネスモデルとの不整合
 - 技術の開示と製品の差別化
- 既存のプラクティスとの不整合
 - 権利化、独占化
 - 内製・外製の二元論
 - 開放型の開発に未対応
- 「オープンソースソフトウェア運動」により衝突の回避、協同が模索されてきた
- 自由ソフトウェアの産業応用のプラクティスの確立は広まりつつある

自由ソフトウェアに対する脅威

- ソフトウェア共同体の消滅
- 自由ソフトウェアの消滅の危機
- 独占ソフトウェアの商習慣の蔓延
- ソフトウェア特許
- DMCA (著作権法の改訂)
- DRM (デジタル制限管理)

- 仕様が非公開のプロトコル
- 仕様が非公開のハードウェア
- 利用に強い制限のあるハードウェア

組込機器応用概観 (1)

- 開発環境として自由ソフトウェアを利用 ☺
- コンパイラ、プログラミングツール
- 実行環境として自由ソフトウェアを利用 ☆
- カーネル、ライブラリ
- 開放型の開発の利点を享受する ☀
- 相互運用性の確保 (通信、フォーマット)
- 既存の資産の活用 (デバイスドライバ)

組込機器応用概観 (2)

- 32-bit 環境は、ほぼサポートされている
- 開発環境として自由ソフトウェアを利用
- 歴史的経緯で POSIX の環境
- GUI で統合開発環境のところは今イチ
- 実行環境は二方面。それぞれ向き不向き
- RTEMS, eCoS, MES などの RTOS
- Linux, NetBSD などを利用
- 経緯と流れ
 - RTOS が自由ソフトウェアになった
 - 既存の自由ソフトウェアが組込機器応用を向いてきた

組込機器応用概観 (3)

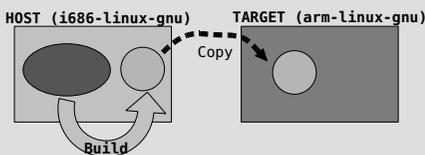
- 開放型の開発の成果の特徴
- よく使われているところは堅い
- 全体としては品質はいろいろ
 - 使えるところを吟味する必要
- 自由ソフトウェアのプラクティスに整合するビジネス形態かどうか注意
- 自由ソフトウェアの理念を拡張する活動
- 内部が自由に研究できるハードウェア
 - OpenSPARC
- 協調開発のハードウェア

GNU 開発環境 GNU Toolchain

- コンパイラ
- GCC (GNU Compiler Collection) 
- アセンブラ、リンカ、etc.
- GNU Binary Utilities (**binutils**)
- デバッガ
- GNU Debugger (**gdb**)
- GNU Make
- GNU autoconf/automake/libtool
- Etc., etc.

組込機器利用の開発 クロス開発

- **host** システム (通常 PC) 上で、**target** システムのために開発する

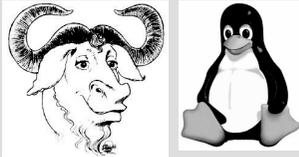


GNU Toolchain の特徴

- 広範な CPU アーキテクチャのサポート :
- x86, ARM, MIPS, PowerPC, SuperH, M32R...
- いろいろな target OS に対応 :
- GNU/Linux, *BSD, RTEMS, ...
- host OS のサポートもいくつか
- (自由ソフトウェアの開発以外にも)
広く使われる実績のある開発環境
- 世界の叡智が集積する開放型の開発

自由の オペレーティングシステム GNU/Linux

- 自由なコンピューティングの基盤を提供
- カーネル、開発環境、さまざまな応用
- 歴史的経緯：構想⇒開発環境⇒カーネル



GNU/Linux の組込機器応用 (1)

- 32-bit CPU では GNU/Linux が大人気
- x86 以外でも多くの挑戦があります：
 - ARM, MIPS, PowerPC, SuperH, M32R, H8/300H, ... (品質はいろいろ)
- 採用には**機器の制約**を考慮する必要がある：
 - CPU の性能
 - メモリの大きさ
 - 入出力インタフェース
 - 電力消費
 - コスト

GNU/Linux の組込機器応用 (2)

- 典型的な利用の構成の例
 - ARM (MMU 付) を CPU に使う
 - フラッシュメモリ と RAM disk をファイルシステムとして利用
 - Ethernet でつながる
 - ブートローダは u-boot
 - BusyBox を利用してコンパクトに

GNU/Linux の組込機器応用に 必要とされる技術 (1)

- ソフトウェア開発の部分
 - Linux カーネルの移植
 - CPU まわり
 - I/O まわり
 - デバイスドライバ
 - ブートローダの移植
 - システムとしてのインテグレーション
 - /etc/init.d のブートシーケンス
 - フラッシュメモリの利用
 - RAM disk の利用
 - BusyBox の利用

GNU/Linux の組込機器応用に 必要とされる技術 (2)

- 通常の GNU/Linux システムに加えて習得すべき技術は、Live Cd の構成の技術とほぼ重なることが多い
- Live Cd の構成の技術があれば、あとは個別のハードウェア対応とブートローダ、とも言える

GNU/Linux の組込機器応用の 勉強に最適な環境

- 玄人指向 KUROBOX/Pro で遊ぼう
- NAS キット
- 2 万円しません
- ハードウェア
 - CPU: 88F5182 (ARM)
 - 128MB RAM / 256MB Flash
 - SATA, USB, Ethernet
- ソフトウェア
 - U-boot, Linux, GNU C library, GNU Bash, Busybox, Samba, ..
- Debian GNU/Linux が動きます

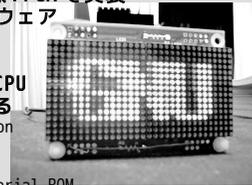


僕等のチャレンジ

- 32-bit での GNU/Linux 利用は進んだ
- だが、デバイス回り、ハードウェア回りでは、まだまだ自由なコンピューティングを阻むものがたくさん
- 自由ソフトウェアの領域を広げよう！
 - 不自由を知る
 - ハードウェア開発に使われる開発環境を自由ソフトウェアに
 - USB 機器などの小規模なところも自由ソフトウェアで

チャレンジ #1 GNU GPL LED Display

- 2006 年トラ技付録 CPLD で実装
- GSoC2007 の課題として DesignWave 付録 FPGA で実装
- 不自由なソフトウェアへの依存を検証
- 作ったもの
特殊用途 RISC CPU
GPLv3 を表示する
 - 8-bit instruction
 - 16-bit data
 - 4-step pipeline
 - ASCII Text in Serial ROM



チャレンジ #2 FSIJ USB Token

- GNU Project 25 周年を祝うデバイス開発
- 当初の目的
 - GNU Privacy Guard の USB トークン
 - 秘密鍵を入れておくデバイス
- USB 汎用 I/O もサポートできるとよい
 - I2C, シリアル入出力, LED 制御, etc.
- FSIJ 会員サービスの認証システムへの応用を期待
- USB 機器開発の回りの自由を検証
- 2008 年 8 月開始

FSIJ USB Token 開発

- 部品の選定とハードウェア設計
 - USB コントローラの選択: ソフトウェアで
 - PCB 設計
- ソフトウェア開発
 - USB プロトコルスタック: AVR-USB を利用
 - タスクライブラリ: Chopstix (今回、開発)
 - CCID/ICCD プロトコルサポート
 - ISO 7816 プロトコルサポート
 - OpenPGP card プロトコルサポート
 - RSA 暗号ルーチン
 - Exptmod, Montgomery-reduction, mul&sq

Atmel AVR CPU

- 自由ソフトウェアに友好的
- 入手製が高く、安価
- 回路の実装が簡単
- ハードアーキテクチャ 8-bit CPU
- GCC がサポートされている
- C ライブラリが整備: AVR-libc
- Simulator: Simulavr
- GDB と Simulavr がつながる
- USBasp ブートローダ
 - 開発時に USB 経由でプログラムを書き込める

開発進捗状況

- “gpg -card-status” 動いた！
- “gpg -clearsign” 動いた！
- 部品そろった
- PCB 設計: 評価第一版できた p-ban.com
- ソフトウェア
 - AVR-USB is ready
 - Task switch&debugging: chopstix ... done
 - ICCD: mostly done (both of on PC and on target)
 - OpenPGP protocol: partially done
 - RSA: mostly done, still buggy
 - Exptmod, Montgomery reduction, mul&sq

まとめ のようなもの

- GNU Toolchain は開放型の開発環境として高い利用価値がある
- GNU/Linux の組込機器利用は進んできた
- コンピューティングの自由の拡大にはやることがたくさん残っている
- FSIJ USB Token 開発、どうですか！
- 自由ソフトウェアを使おう！
- 開放型の開発に参加しよう！

Happy Hacking!