次世代GRUBブートローダの基本設計と実装

国際フォーラム篇

奥地　秀則 <okuji@enbug.org>

発表の流れ

- プロジェクトの概要
- サブプロジェクトの説明
- プロジェクトの背景や目的
- 現状報告
- デモ
- 今後の展望

概要

GNU GRUBブートローダの次世代バージョンの基盤を開発する。

- コードの浄化。
- 安全性の向上。
- 拡張性の向上。
- 移植性の向上。
- 国際化のような、従来では実現困難であった機能の追加。
- 有用であれば、それ以外のソフトウェアも開発。

コードネームは、PUPA。

サブプロジェクト: BugCommunicator

- Rubyで書かれた**バグ追跡システム**（BTS）。
- GRUBで使っていたSavannahのCodeXが腐っているので、開発。
- Debian BTSに近いが、拡張性に富む。
- 最新リリースは0.3。
- ホームページは
  <http://www.nongnu.org/pupa/bugcomm.html>。
- <http://bugcomm.enbug.org/>にて、すでに運用中。

サブプロジェクト: Ruby/Cache

- LRUアルゴリズムに基いて、オブジェクトをキャッシュするための、Ruby用ライブラリ。
- BugCommunicatorのレスポンスを高速化するために開発。
- しかし、使わなくても案外速かったので、実はまだ使っていない。
- 最新リリースは0.3。
- ホームページは
  <http://www.nongnu.org/pupa/ruby-cache.html>。
- 新井康司氏のFileCacheをサンプルとして添付。
- まつもとゆきひろ氏の反応がいまいちなので、現在は開発停止。

GRUBの紹介

- IBM PC用に作られた高機能なブートローダ。
- Multiboot Specificationに準拠し、Linuxや*BSDは直接ブート可能。その他はチェインロードで対応。
- メニューやコマンドラインを用いたインターフェース。
- 数多くのファイルシステムに対応。
- シリアル端末やherculesコンソールに対応。
- ネットワーク対応。
- ディスクレス・システム対応。

おかげで、多くのGNU/LinuxディストリビューションやOS開発プロジェクトに採用され、組み込み機器や大規模PCクラスタにも普及した。**が...**

GRUBの不運

1. 元々はそんなにたくさんの機能はなかった。
2. 新しい機能に関する要望が続出。
3. 当然、場当たり的な解決が頻発。
4. 暇を見付けて、かなり書き直したが、
   想定されていなかった機能の追加が後を絶たず。
5. スパゲッティの出来上がり。

GRUBに寄せられる無茶な願望（一部）

設定ファイルを分割したい

一度に一つしかファイルを開けませぬ。

変数やif文が使いたい

メモリ管理が原始的だから無理です。

Powerで動かないの？

i386ベッタリなコードで書かれてます。

パーティションを消したら動かない

Stage2はファイルシステムに置かれてます。

機能を追加し難い

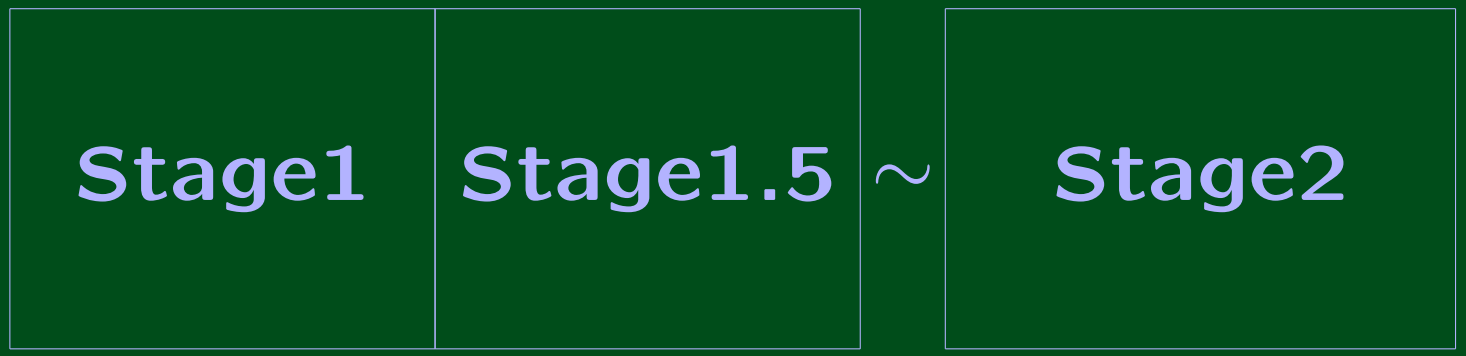
ちゃんと構造化しないとね...

プロジェクトの目的

- 救援モードによる安全確保。
- 動的リンクによる拡張機能。
- 真のメモリ管理機能。
- 構造化されたフレームワーク。
- CPUやマシンに依存したコードの明確な切り分け。
- 異なるアーキテクチャ上でのインストール機能。
- 過去の汚物はばっさり。

従来のGRUBの配置

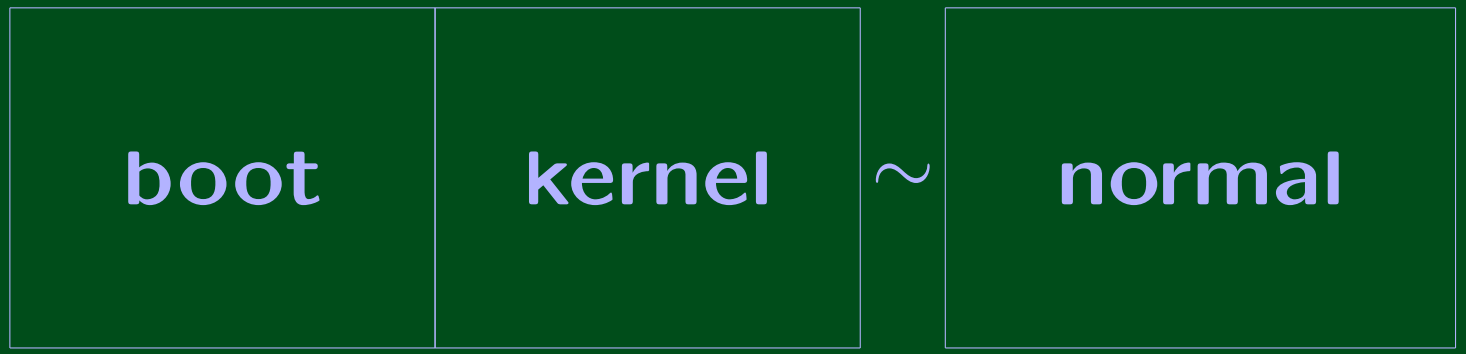
典型的な場合、Stage2はファイルシステムに置かれる。



Stage2がロードできなければ、GRUBは**不能**になる。

PUPAの配置

空き領域(通常62セクタ)を有効利用する。



仮に通常モードがロードできなくても、**救援モードで復旧**できる。

PUPAの構成

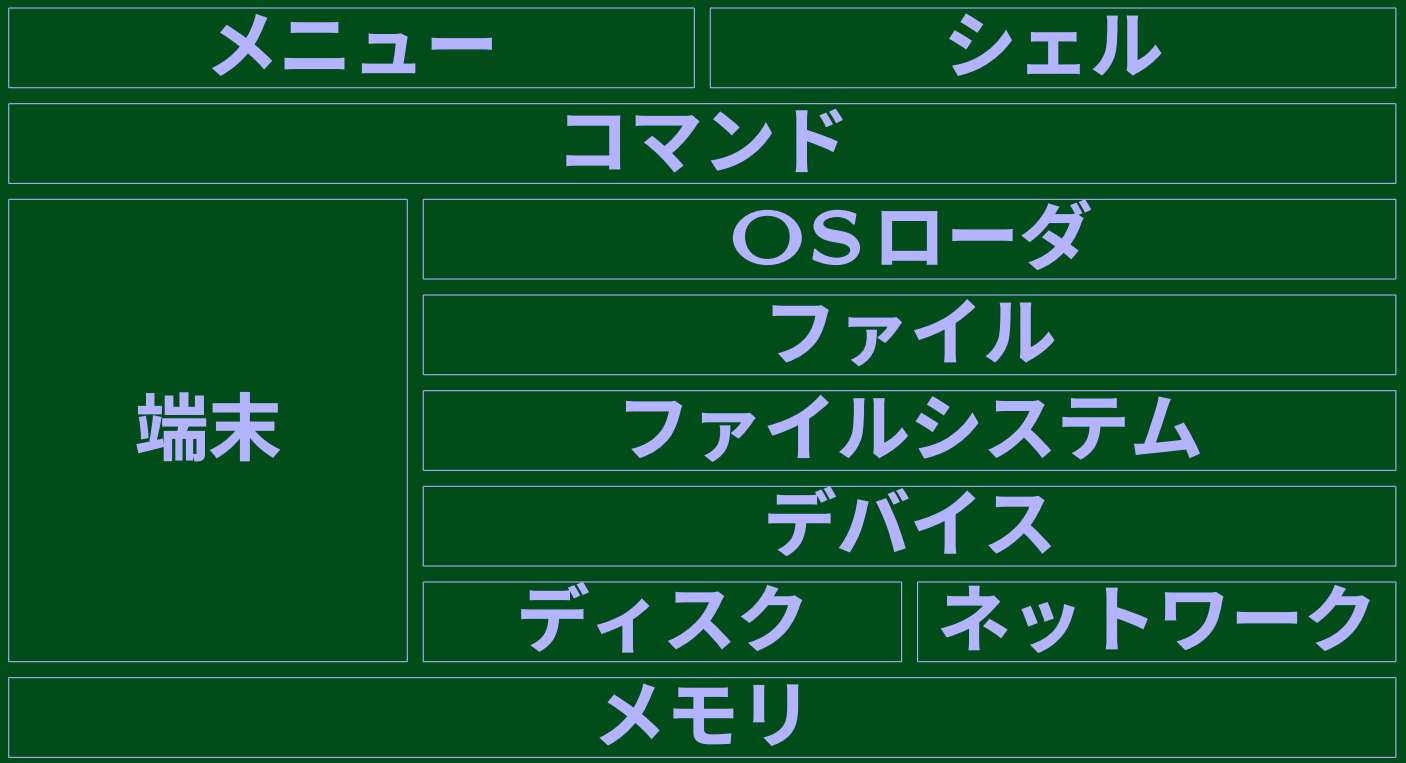
- boot、diskboot、kernelの三つのイメージと
  モジュール群から成る。
- モジュールはELFのrelocatable objectを利用する。
- kernelは実行時にモジュールをロードし、
  自分自身に機能を追加する。
- モジュールのロードに必要な機能は
  前もってkernelにロード(preload)しておく。

PUPAの構成の利点

- **再コンパイルの必要が無い**ので、
  異なる構成のkernelを容易に利用できる。
- コード中の場合分けが無いので、**コード管理が簡単**である。
- モジュールの**独立性を高める**ことができる。

PUPAのフレームワーク

OSのような構造化されたフレームワークによって、
管理や拡張を容易に。



コードの浄化

- GRUBでは、Makefileの生成に
  Automakeを半ば強引に利用している。
- PUPAでは、Rubyで記述された
  独自のMakefileジェネレータを作成して利用。
- 汚いコードは、i386への依存性の排除を含めて、
  徹底的に書き直し。
- 言葉で説明するのは難しいので、ソースを読んでください。

現在の状況（1）

ブートストラップ 完了。

　サイズの問題をもう少し考える必要があるが。

メモリ管理 完了。

動的リンク 完了。

フレームワーク 完了。

救援モード 大体完了。

通常モード 完成にはまだ遠い。

現在の状況（2）

ファイルシステム FATのみ。

ディスク サポート済み。

ネットワーク 今回は見送り。

端末 コンソールのみ。

OSローダ チェインロードとLinuxに対応済み。

インストーラ 不完全だが、動作する。

PowerPCからでもインストール可能なことは検証済み。

国際化 全く進んでいない(共同開発者の松嶋の担当)。

デモ

動く所をBochs IA-32 Emulatorを使って見せる。

1. 通常モードが動いているところを見せる。
   救援モードに入ったり、通常モードに戻れることを示す。
2. 通常モードがロードできない場合に、
   救援モードに突入するところを見せる。
3. 救援モードからLinuxをロードするところを見せる。
   完全にブートさせると時間がかかるので、途中でやめる。

さらなる情報

http://www.nongnu.org/pupa/

これからの予定

残りの期間で以下の事を行う。

- 空き領域に埋め込むにはサイズが厳しくなってきたので、自己解凍を使ってイメージを縮小する道を模索する。
- 通常モードを使えるレベルにする。
- 全然進んでいない国際化関連を何とかする。

このプロジェクトが終わった後の予定

- PUPAという名称は捨て、GRUB 2として開発を続行する。
  現行のGRUBを駆逐するには後一年はかかるかな？

- 本当に移植できるのか、何かで試してみたい。

聞いてくれてありがとう!