

RubyによるOS構築法の研究

吉原 陽香* 笹田 耕一† 佐藤 未来子‡ 並木 美太郎§

* 東京農工大学大学院工学府情報工学専攻 † 東京大学大学院情報理工学系研究科創造情報学専攻

‡ 東京農工大学大学院工学府 § 東京農工大学大学院工学研究院

1 はじめに

従来OSの記述に使われてきた言語Cは、ハードウェア制御が行いやすい一方、不正なメモリアクセスやメモリリークなどのプログラマから発見しにくいバグを生み出す可能性がある。対してスクリプト言語はインタプリタやVMで実行され、上記のようなバグを未然に防ぐことができる。そのスクリプト言語の1つであるRuby[1]は、まつもとゆきひろらによって開発されたオブジェクト指向のプログラミング言語である。

そこで本研究は、Rubyを用いてOSを構築することを目的とした。計算機資源をRubyのオブジェクトに仮想化し、見通しのよい構成をもつOSによる資源管理をRubyにて実装することを本研究の目的とする。

2 関連研究

ここでは、OSの搭載されていないハードウェア（以下、ベアマシン）上でC以外の言語を直接実行した例を示す。

M3Lマシン[2]は、ある特定のハードウェア上に対応するマイクロプログラミング言語を用いてLISPのインタプリタをハードウェア上に実装している。この研究のように、専用のハードウェア上でしか動作しない処理系を実装すると移植性が損なわれてしまう。

OCOS[3]とは、関数型言語MLの方言であるOCamlにて記述したOSである。ハードウェア上にてCやアセンブラで記述されたランタイムシステムと、OCamlから実メモリやI/OポートにアクセスするためのI/Oアクセス層が動作し、OCOSはそれらを利用して動作する。本研究の場合、Cにて拡張ライブラリを実装することで実メモリやI/OポートへのアクセスがRubyにて記述可能になると考えている。

A Study of Operating System Construction with Programming Language Ruby

* Haruka YOSHIHARA †Koichi SASADA ‡Mikiko SATO §Mitaro NAMIKI

* Faculty of Engineering, Tokyo University of Agriculture and Technology.

†Department of Creative Informatics, Graduate School of Information Science and Technology, The University of Tokyo.

‡Faculty of Engineering, Tokyo University of Agriculture and Technology.

§Institute of Symbiotic Science and Technology, The Graduate School at Tokyo University of Agriculture and Technology.

3 目標

本研究では、OSの行うプロセス・スレッド管理、メモリ管理、ファイルシステム、デバイスドライバ、割り込み処理をRubyにて記述することを目標とする。OSやAPはRuby処理系内に組み込まれたVMにて実行する。メモリリークなどの問題を回避し、また計算機資源を仮想化してRubyのオブジェクトとして管理することで、安全性や生産性、拡張性に優れ、統一の見通しのよい構成をもつOSを目指す。またターゲットハードウェアはx86とし、周辺機器としてキーボード、マウス、ディスク、ディスプレイを扱う。

4 設計

RubyOSの構成を図1に示す。

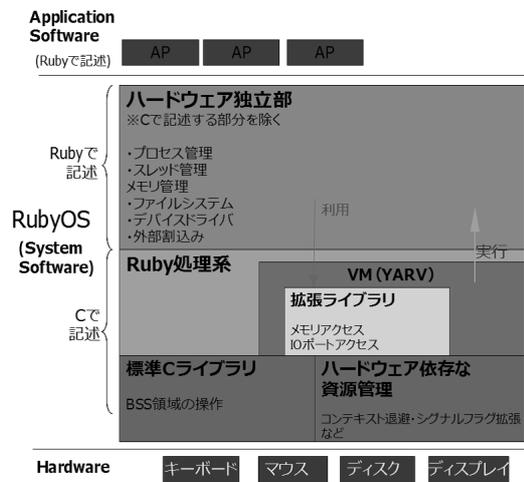


図1: RubyOSの設計

計算機資源をオブジェクトに仮想化し、Rubyにてその管理を行う処理を記述する。またRuby処理系にはCで記述された公式の処理系を利用し、組み込み用Cライブラリを利用してベアマシン上で動作させる。

4.1 プロセス・スレッド管理

プロセスは、「Rubyで記述されたAPを実行するRubyのVM」として定義する。またOSコードもRubyのVMにて実行し、プロセスのスケジューリングを行う。スレッドの実装はRubyのFiber[4]クラスを利用する。Fiberとはノンプリエンティブなユーザレベルのスレッド

であり、RubyOS では Fiber 用のスケジューラを作成しスレッドを実装する。

4.2 メモリ管理

メモリ管理はプロセスに割り当てる物理メモリの割り当てや解放を行う。プロセスや OS のメモリ使用状況が競合しないように、OS と全プロセスそれぞれに異なるアドレス空間を割り当てる。

4.3 データの永続化 (ファイルシステム)

RubyOS ではオブジェクトをシリアライズ化したものをディスク上に格納し、オブジェクトを永続化する。データの永続化には Ruby のライブラリである PStore を使用する。PStore はハッシュの形式を用い、オブジェクトをシリアライズしたデータに対しキーを割り当てて保存するライブラリである。RubyOS ではこの PStore を修正したものをを用いて Ruby のオブジェクトをディスク上に保存する。

4.4 デバイスドライバ・割り込み制御

デバイスドライバはそのデバイスごとにクラスとして Ruby で記述し、それぞれ仮想化する。外部割り込みの処理に関しては Ruby 処理系に実装されているシグナルチェックの機能を利用する。

4.5 記述言語の検討

RubyOS ではすべての資源管理機能を Ruby にて記述することを目的としているが、コンテキストスイッチのためのレジスタ退避など、Ruby 処理系の動作を中断させるような動作は Ruby を用いて記述することはできない。これらについては C かアセンブラを利用して記述する。また、メモリや I/O ポートへのアクセスに関しては、C を利用して拡張ライブラリを記述し、Ruby 処理系を拡張する。

5 実装

RubyOS の実現環境は PC/AT 互換機 (Intel celeron 2.66GHz, 1.25GB) であり、Ruby のバージョンは 1.9.1-p129 である。現在までに、Ruby 処理系の最小単位である miniruby をベアマシン上へ移植した。

またメモリ・I/O ポートアクセスを Ruby から行うための拡張ライブラリを作成し、キーボードとテキスト VRAM のドライバを Ruby にて実装した。この 2 つのドライバを利用して記述した、入力された文字を Ruby スクリプトとして実行するスクリプトを QEMU 上にてエミュレートした様子を図 2 に示す。

今回 Ruby にて作成したこのドライバを活用する例として、PIO による ATA デバイスの読み込みを行い、

```

QEMU
Hello Ruby VRAM & KEYBOARD!!!
[1,"abc",[3,4]].each{|x|
  p x
}
puts "end"
EOR
-----
1
"abc"
[3, 4]
end

```

図 2: スクリプトの実行結果

値を 16 進法で表示する Ruby スクリプトを記述した。その際、QEMU で実行するためイメージファイルを利用し、このファイルに Unix 上で hexdump コマンドを用いて、Ruby スクリプトで表示した値が正しいことを確認した。また実機のベアマシンでも動作を確認した。なお、現在 RubyOS 上で走行している VM は 1 つであり、実メモリ空間にて実行している。

上記の結果から、ベアマシン上に Ruby の実行基盤を実装することができた。さらにキーボードとテキスト VRAM のドライバを Ruby にて記述し、Ruby のオブジェクトに仮想化した。また Ruby でのディスクアクセスを確認した。なお、現在までに移植のために記述した C のソースコードの行数は 1486 行であり、Ruby で記述した OS 部分は 250 行である。

6 おわりに

本報告では、Ruby によって記述する OS の構築法とこれまでに行った実装について述べた。計算機資源を Ruby オブジェクトに仮想化することを設計指針とし、それに則り実装を行うことで、見通しの良い OS コードになることを確認した。今後の課題として、まずはオブジェクトの永続化機能を実装する。

参考文献

- [1] まつもとゆきひろ, 石塚圭樹. オブジェクト指向スクリプト言語 Ruby. 株式会社アスキー, 1999.
- [2] J. P. Sansonnet, M. Castan, C. Percebois, and J. Perez D. Botella. Direct execution of lisp on a list-directed architecture. In *Architectural Support for Programming Languages and Operating Systems(Proc. ASPLOS-I)*, pp. 132–139. ACM, mar 1982.
- [3] 井上翔大, 大山恵弘. Ocaml による os の実装. 情報処理学会「システムソフトウェアとオペレーティング・システム」研究会第 113 回研究報告, Vol. 2010-OS-113, No. 4, Jan 2010.
- [4] 芝哲史, 笹田耕一. Ruby1.9 での高速な fiber の実装. 第 51 回プログラミングシンポジウム予稿集, pp. 21–28, 2010.