

# OS 起動の仕組みをアセンブリ言語レベルで学ぶためのシミュレータ教材の開発

玉野 太策, 中西 通雄

Taisaku Tamano, Michio Nakanishi

大阪工業大学情報科学部コンピュータ科学科

Department of Computer Science, Faculty of Information Science and Technology,

Osaka Institute of Technology

Email: michio.nakanishi@oit.ac.jp

あらまし：情報を専門とする大学生を対象として、OS 起動の仕組みを学習することを目的としたシミュレータ教材を開発した。シミュレータは、x86 アーキテクチャの命令セットに対応する CPU エミュレータを搭載しており、命令をステップ実行しながらレジスタやメモリの値を確認することができる。また、外部記憶装置の内容を 16 進表示することにより、パーティション情報やブートシグネチャといった情報を確認可能にした。用意した印刷教材に沿ってシミュレータを用いながら、OS が起動するまでの仕組みをアセンブリ言語レベルで学習することで、CPU とメモリ、外部記憶装置との関係や HDD の構造の知識、実用的なアセンブリ言語の使用例を理解することができる。

キーワード：ブートストラップ, OS, アセンブリ言語, シミュレータ

## 1. はじめに

パソコンは電源が入れられると、まず初めに基板上の ROM から BIOS の命令列がメモリに読み込まれる。この命令列は、外部記憶装置内の Master Boot Record (MBR) のブートシグネチャを確認して起動可能であるかを確認する。起動可能な外部記憶装置であれば、その外部記憶装置の MBR の内容をメモリに読み込み、プログラムカウンタの値を、読み込んだメモリ領域の先頭番地へと変更する。外部記憶装置が HDD の場合、MBR にはブートローダのプログラムと HDD のパーティション情報が記載されている。ブートローダはパーティション情報を読み取って OS の含まれるパーティションを探し、OS をメモリに読み込む。以上の流れを Bootstrap という。

本研究では、アセンブリ言語の基礎知識を有する大学生を対象として、上記のような OS の起動の仕組みの学習を支援するシミュレータ教材を開発した。本シミュレータでは、OS 起動までに実行される機械語の内容をアセンブリ言語で表示し、CPU のレジスタの値やメモリの値などの変化をリアルタイムで確認できる。また、外部記憶の内容を 16 進数で表示することよりのパーティション情報やブートシグネチャといった情報を確認することを可能にしている。本シミュレータを授業内で使用することにより学習者の理解度の向上を図る。

## 2. 実装方法と動作環境

### 2.1 実装方法

本シミュレータには、x86 アーキテクチャの機能を必要最低限の範囲で実装したエミュレータを搭載している。このエミュレータは、市販に付属しているエミュレータのプログラムを参考に、学習に必要な機能を追加して実装した<sup>(1)</sup>。グラフィカルイン

ターフェースの実装には、クロスプラットフォームアプリケーションフレームワーク Qt5 を用いた。これらの実装には C/C++ を用いており、開発したソースコードは約 1500 行である。また、シミュレータ全体の容量は約 120KB である。また、シミュレータ内に動く BOIS やブートローダのプログラムには、フリーソフトのアセンブリ言語 NASM を用いている。

### 2.2 動作環境

本シミュレータは Ubuntu14. 04LTS 以上で動作する。また、エミュレータ内の文字入出力をターミナル上で行うため、本シミュレータの起動はターミナルより行う必要がある。

## 3. 仕様と概要及び利用方法

### 3.1 画面仕様

本シミュレータを起動すると、図 1 の画面が表示される。画面サイズは 1340\*710 ピクセルである。画面は 6 つの部分で構成されており、その概要は表 1 のとおりである。

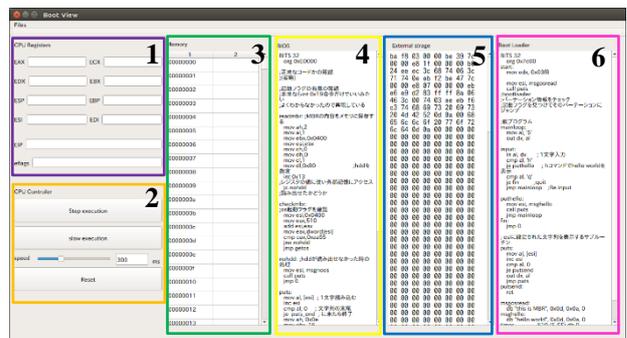


図 1 画面の全体

表 1 各部の名称と内容

位置	名称	内容
1	Register 群	汎用レジスタの値 プログラムカウンタの値 フラグレジスタの値
2	CPU Controller	命令の制御, CPU のリセットを行う
3	Memory	メモリのアドレスと値
4	BIOS	BIOS 内の命令をアセンブリ言語で示したもの
5	External Storage	HDD 内の情報を 16 進数で示したもの
6	Boot Loader	ブートローダをアセンブリ言語で示したもの

### 3.2 機能概要

搭載されているエミュレータは, CPU Controller の Step execution 又は Slow execution のボタンを押すことにより, 図 1 の BIOS 部に表示された命令を実行する. BIOS には外部記憶装置のブートシグニチャを確認し, その内容をメモリに読み込むように命令が設定されている. 本シミュレータと同じディレクトリに置かれている storage.ves ファイルは外部記憶装置の代わりとしており, ブートローダのプログラムと OS の代わりとして自作した簡易的なプログラム「Prov OS」が書きこまれている. Prov OS が読み込まれると, ターミナル画面が開くので, 学習者はブートストラップが成功したことを確認できる. Prov OS は, h が入力されたら Hello World を出力し, q が入力されたら終了する. ターミナル上では図 2 のように表示される.

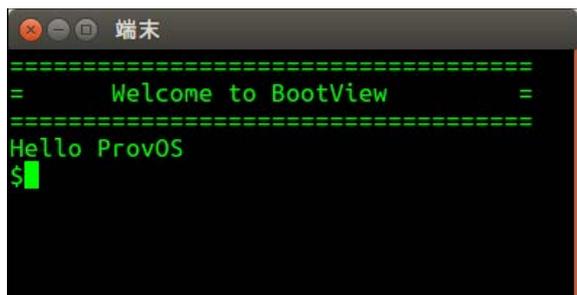


図 2 OS の代わりのプログラム実行画面

### 3.3 利用方法

学習者は, ボタン操作を行うことにより, CPU が命令を実行していく様子を確認することができる. 実際に BIOS 内の命令が実行されている様子が図 3 である.

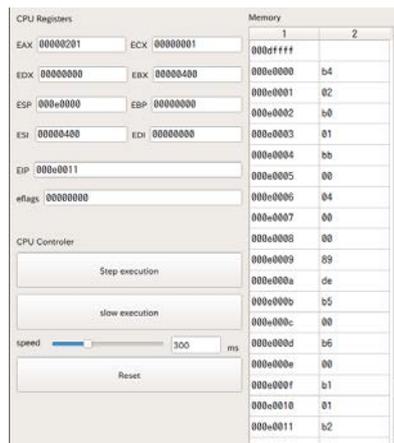


図 3 シミュレート時の画面

## 4. 教材について

本シミュレータは, 印刷教材と併用して利用することを想定しており, 印刷教材では次のことを解説する.

- (1) x86 アーキテクチャの汎用レジスタの名称や各レジスタの使用方法
- (2) HDD のシリンダ, セクタ, トラック
- (3) BIOS に書きこまれている命令列
- (4) ブートローダとパーティション情報

学習者は, 印刷教材で解説されたプログラムの内容を本シミュレータを利用することにより確認することができる.

## 5. 評価について

評価方法として, はじめに事前テストを行う. 次に教材の流れ従って授業を行う. このとき, 学習者にはシミュレータを使用してもらい, その後に理解度テストを行う. 理解度テストでは CPU とハードウェアの関係や Bootstrap についての理解度を尋ねる. また, 学習者がシミュレータを用いたことにより, 理解度の向上につながったかについてのアンケートも行う予定である.

## 6. 結論

本研究では, OS 起動の仕組みをアセンブリ言語レベルで学ぶためのシミュレータ教材の開発を行った. シミュレータを使用することにより, 学習者が OS 起動の仕組みを学ぶ上での理解度向上の助けになると考えている.

本校の執筆時点では印刷教材は製作途中であるため, 本シミュレータを試用してもらっていない. 評価結果については研究発表のときに口頭にて報告したい.

### 参考文献

- (1) 内田公太, 上川大介:「自作エミュレータで学ぶ x86 アーキテクチャ」, マイナビ (2015.8)