自由な PC 向けファームウェアの開発
—yabits: Yet Another UEFI Implementation—

1. 背景

現在の汎用のコンピュータ（PC）には Basic Input Output System（BIOS）と呼ばれるソフトウェアが内蔵されている. BIOS は PC が起動するときに最初に実行されるソフトウェアであり，ハードウェアの初期化や Operating System（OS）へのサービスの提供などが主な役割となっている.

一般的な x86 系の PC の場合，BIOS には大きく分けて Legacy BIOS と Unified Extensible Firmware Interface（UEFI）の 2 種類が存在する. Legacy BIOS は 1981 年に製造された IBM PC の BIOS を元に拡張が繰り返された BIOS であり，後方互換性維持のための制約や，標準化がなされていないなどの問題を抱えている. こうした Legacy BIOS の問題を解決すべく設計されたのが UEFI である. UEFI は UEFI Forum により標準化されており，IA-32 や RISC-V など様々なアーキテクチャをサポートしているなど非常に柔軟な設計となっている.

現在，UEFI の実装は Intel 主導によりオープンソースで開発が進められている TianoCore とそれを元に各 BIOS ベンダが独自の機能を追加したプロプライエタリな UEFI 実装の 2 種類が存在する.

BIOS は通常，後者の BIOS ベンダによりプロプライエタリなソフトウェアとして提供されており，ユーザに対してはそのソースコードが公開されていない. そのため，ユーザがファームウェアを修正したいと思っても容易ではなく，リバースエンジニアリングなどを行い解析しないと修正ができないという問題がある. これは深刻なバグが放置されたり，コードが十分に検証されないまま利用されたりするなどの問題にも繋がる. こうしたことから，既存のファームウェアをオープンソースな実装に置き換えようとする動きがある.

coreboot はオープンソースなファームウェアを開発するプロジェクトの代表格であり，Chromebook などでも採用されている実績がある. coreoboot はマシン依存部となる coreboot 本体と payload の 2 つの部分から構成される. payload には様々なソフトウェアを搭載可能となっている. 例えば，先に説明した TianoCore も coreboot payload として搭載することができる仕様となっている.

しかし，coreboot payload としての TianoCore は OS の起動という面から見た場合，TianoCore 自身が不必要な機能を数多く持っていることから，フットプリントの増大や起動時間の増加などの問題を抱えている.

2. 目的

以上を踏まえて，次の条件を持った UEFI 実装が求められていると言える.

- 高速
- 軽量
- オープンソース
- TianoCore とコード非共有

本プロジェクトでは，これら 4 点の条件を満たす UEFI 実装を開発した．この UEFI 実装により，TianoCore を代替することを目標とし，既存の UEFI 実装に存在する問題の解決を目的とした．

3. 開発の内容

本プロジェクトでは OS の起動にのみフォーカスした必要最小限の機能を持った UEFI 実装 yabits/uefi（以下，yabits）を開発した．

yabits は coreboot payload として実装されているため，coreboot がサポートしていれば QEMU などの仮想マシンを含め，どのようなマシンでも動作する．実際に yabits は QEMU や Lenovo ThinkPad X230 で動作することが確認されている．

OS を起動させるためには，通常の UEFI と同等のインターフェースを持っている必要がある．このため，yabits は UEFI の規格に合わせたインターフェースを持っており，bootloader や OS はこれらのインターフェースを介して各種機能を利用することが可能となっている．実装されている機能としては，BootServices や RuntimeServices，EFI System Table 対応や GUID Partition Table Disk Layout 対応などがある．一方で今後は UEFI に取って代わられるであろう Legacy BIOS との互換性は持たせておらず，純粋な UEFI として実装されている．

yabits は既に無修正の各種 UEFI アプリケーションや，図 1 に示すように代表的なオープンソースの bootloader である GRUB2 に加え，Linux や OpenBSD が起動するほどの高い互換性を持っている．

BIOS は Pre EFI Initialization（PEI）フェーズに始まり OS の起動する Run Time（RT）フェーズまで，様々なフェーズに移行しながら起動を進める．これらのフェーズのうち，Driver Execution Environment（DXE）フェーズではドライバの読み込みを行う．TianoCore では DXE フェーズで読み込まれるドライバの数が多く，起動時間が長くなるなどの問題があった．

本システムでは，IDE や AHCI などの OS の起動に必要最小限のデバイスドライバのみを実装している．これにより，図 2 に示すように TianoCore と比較して DXE フェーズでのドライバの読み込みを少なくし，起動時間の短縮とフットプリントの縮小を図っている．

図 1 yabits から起動された GRUB2

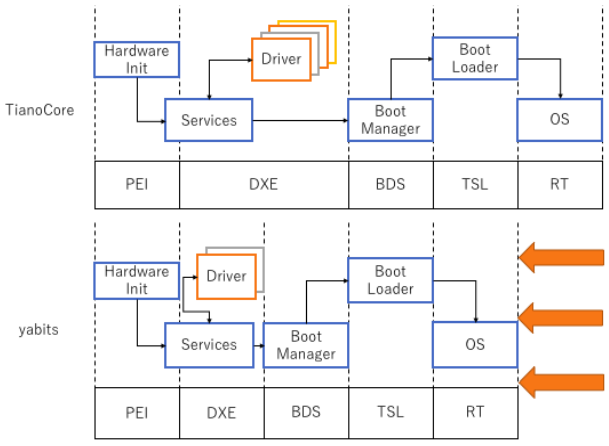

図 2 yabits と TianoCore との起動フェーズの比較

4. 従来の技術(または機能)との相違

背景で説明したように，BIOS ベンダにより提供される UEFI 実装はバグが修正されずに放置されたり，クローズドで開発されているためにコードが十分に検証されないまま利用されたりするなどの問題がある．一方，TianoCore は OS の起動には不要な機能を多くロードするため，フットプリントが増大し起動時間が長くなるなどの問題がある．

本システムは OS の起動に必要最小限の機能のみを提供するため，表 1 に示すように，既存の UEFI 実装と比較してフットプリントが小さく，かつ起動時間が短いなどの利点をもっている．また，目的で挙げた 4 つの条件について，他の UEFI 実装との比較は表 2 のようになり，本システムのみがこれらの条件を全て満たすことがわかる．

表 1 yabits と TianoCore との比較

	起動時間(秒)	フットプリント(MB)
yabits	3.15	0.4
TianoCore	7.15	4.1

表 2 yabits と他 UEFI 実装の比較

	起動時間	フットプリント	オープン性	TianoCore とのコード非共有性
yabits	○	○	○	○
既存の UEFI	×	×	×	×
TianoCore	×	×	○	×
NERF	○	○	○	×

5. 期待される効果

ハードウェアがリセットされてから OS が起動するまでを担うという点においては既存のファームウェアと大きな差はない．一方で，既存の UEFI 実装と比較してフットプリントが小さいことから，一般的な PC のみならず組み込みデバイスなどでの利用も考えられる．また，起動時間が短いことから，ベアメタルクラウドにおいてインスタンスを起動した際の起動時間の短縮やコストカットが見込まれる．

近年のオープンソース化の流れにより，これまでクローズドだったソフトウェアがオープンソースとなり，容易にユーザがそのソースコードを閲覧，編集できるようになった．本システムにより，これまであまり考慮されることの少なかったファームウェアなどの分野においてもオープンソース化の流れが一層強まることが期待される．

6. 普及(または活用)の見通し

今後は本システムの周知を目的として国内外のカンファレンスでの発表や，Web 上での活動を行い，開発コミュニティの形成に努める．また，BIOS ベンダに積極的にアプローチし，デフォルトの BIOS として採用されることを目指す．

7.　クリエータ名（所属）

師尾　彬 （東京理科大学大学院）

（参考）関連 URL

https://github.com/yabits/uefi