

GearsOS の Boot に関する研究

175733E 氏名 奥田光希

指導教員：河野 真治

Abstract

We are proposing GearsOS, which uses Meta-calculus to improve reliability in our lab. We are implementing GearsOS on Raspberry Pi based on xv6. Currently, GearsOS is booted from BIOS, and we would like to migrate it to UEFI. If we can migrate to UEFI, we can avoid dependencies on CPU and other devices. GearsOS is written in CbC (Continuation based C), so it is not affected by CPU or device. This will make it possible to apply GearsOS to a variety of embedded systems.

1 はじめ

2017 年、Intel 社は 2020 年までに Legacy BIOS と UEFI への互換を非推奨とし、互換モジュールの CSM を削除すると発表した。[1] Legacy BIOS は PC の進化に伴い、致命的な問題点が発生する。問題点の一つとして、拡張性がないことがあげられる。Ethernet や USB につながるディスクなど、新たにブートデバイスが追加されるたびに、OS のブートローダを変更しなければならない。またマザーボードごとに、ファームウェアをアセンブラで開発する必要がある。また、1MB のメモリ制限により、セキュリティを含めたシステム機能の強化が困難であるためセキュリティにも問題がある。これらの問題を解決するために UEFI が開発された。UEFI は、2TB を超える大きなディスクからブートでき、高速にブートできる。CPU に依存しないアーキテクチャとドライバを持ちネットワークも使用可能な柔軟なプレ OS 環境が利用できる。今後、Legacy BIOS から UEFI への移行が急速に進むだろう。また、UEFI は Application を持っていて、それは C 言語などの高級言語で記述可能なので個人で開発がしやすい。

当研究室では、信頼性と拡張性をテーマに GearsOS を開発している。GearsOS は Continuation based C(CbC) によってアプリケーションと OS そのものを記述している。現在、CbC で証明可能な OS を実装するために、xv6 の CbC の書き換えを行っている。xv6 は Legacy OS のため、UEFI から起動することができない。UEFI から xv6 を起動させることができれば、拡張性が大きく広がる。さらに、UEFI は接続されているデバイスの認識を持てるので、マウスやキーボードを使えるように UEFI Application を使い、デバイスドライバを作れる。これにより、利便性もよくなる。本研究では、ARM で動くシングルコンピュータである Raspberry Pi 上に UEFI から GearsOS をブートさせることにより様々な組み込みシステムに対して GearsOS を応用できる様にするのである。

2 Continuation based C(CbC)

Continuation based C(CbC)[2] は、当研究室で開発を行っているプログラミング言語である。CbC は、C 言語の下位言語であり、関数呼び出しではなく継続を導入している。CbC では、関数の代わりに CodeGear という単位でプログラミングを行う。CodeGear は入力と出力を持ち、CbC では引数が入出力になっている。CodeGear から次の CodeGear へと goto による継続で遷移して処理を行い、引数として出力を与える。

2.1 CbC GCC ARM CrossCompile

GCC 上で実装された CbC の ARM[4] CrossCompile[3] 環境を Singularity[5] で構築した。CrossCompile は、Compiler が動作している以外のプラットフォーム向けに実行ファイル生成する機能を持った Compile 手法である。これにより、CbC で実装されたプログラムを Raspberry Pi 上で実装ができるようになった。

3 Gears OS

GearsOS[6] は当研究室で信頼性と拡張性をテーマに開発を行っている OS である。GearsOS は Continuation based C(CbC) によって記述されている。当研究室では、GearsOS の信頼性と CbC の有効性を示すために、基本的な機能を揃えた OS である xv6[7] を CbC で置き換えを行っている。これにより、OS の個々のシステムコールが持つ状態を明確にすることができると考えている。CbC で書き換えられた xv6 を Raspberry Pi[8] に搭載することでハードウェア上でのメタレベルの計算や並列実行を行える様になる。

4 UEFI

UEFI[9] は、OS とプラットフォームファームウェアの間のソフトウェアインタフェースを定義する仕様である。UEFI

は、インタフェース仕様であるため、特定のプロセッサに依存しない。C 言語などの高級言語で実装することができ、EDK2[10] や gnu-efi[11] などの UEFI ツールキットがあり個人でも開発の難易度が低い。

UEFI には Application があり、UEFI Application は開発を行うマシンの OS やアーキテクチャに左右されることなく、高級言語から直接生成することができる。UEFI Application を自作することにより、xv6 を UEFI で Boot させる BootLoader を作ったり、xv6 のデバイスドライバを作れるようになる。

5 BootLoader

BootLoader は OS を Boot させるプログラムである。BootLoader が OS を Boot するまでのプロセスを図 1 に示した。UEFI が起動し、Boot Manager が Boot Loader を起動させる。BootLoader が OS の Kernel をメモリに Load し、Kernel が init プロセスを起動させる。init プロセスが OS の Boot プロセスを起動させ、OS が Boot される。

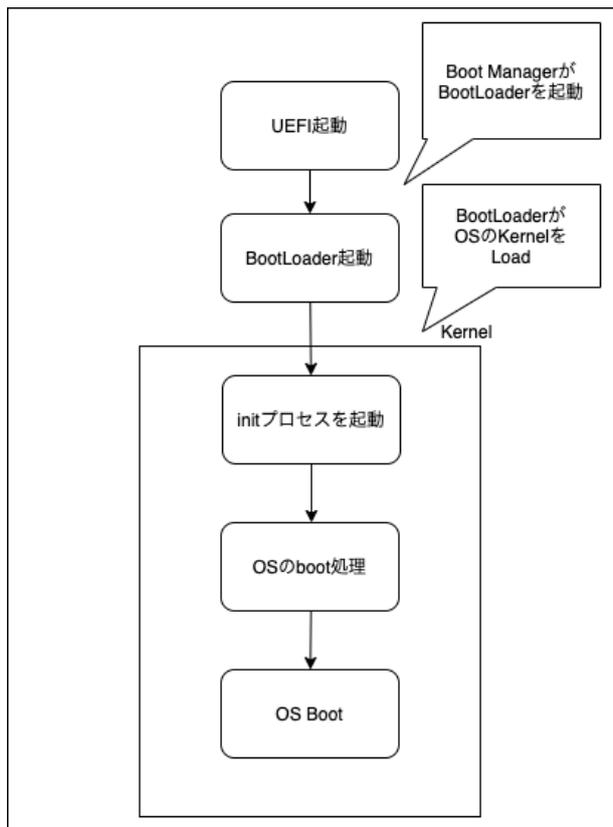


図 1: Boot Loader process

6 研究の成果

- UEFI の開発環境を Singularity で作成した
- gnu-efi で作成した UEFI Application を qemu-system-arm で動かした

- Raspberry Pi に UEFI のファームウェアとして設定し、実行した
- ミニマムな Kernel Loader を調査し ARM xv6 用に書き直した

7 今後の課題

- Singularity 上の qemu-system-arm に gdb を接続する
- その gdb で Kernel Loader をバックする
- xv6 Kernel を UEFI から Boot するコードを入れる
- xv6 を書き換えた GearsOS を実装する
- USB Driver を実装し、キーボードやマウスを使える様にする

参考文献

- [1] Intel/Unified EFI Forum, <https://www.uefi.org>,2017/11/3.
- [2] 清水隆博, 河野慎治..xv6 の構成要素の継続の分析. 情報処理学会研究報告,2020.
- [3] https://wiki.osdev.org/GCC_Cross-Compiler
- [4] <https://developer.arm.com>
- [5] <https://sylabs.io/singularity>
- [6] 東恩納 琢偉, 河野真治.Gears OS でモデル検査を実現する手法について. 情報処理学会研究報告 2020
- [7] Russ Cox, M Frans Kaashoek, and Robert Morris. Xv6, a simple unix-like teaching operating system,2012.
- [8] <https://www.raspberrypi.org>
- [9] <https://wiki.osdev.org/UEFI>
- [10] <https://github.com/tianocore>
- [11] <https://sourceforge.net/projects/gnu-efi>