

令和2年度 卒業論文

Gears OS Device Driver



琉球大学工学部工学科知能情報コース

175701G 氏名 奥田 光希

指導教員：河野 真治

目次

第 1 章	初めに	1
1.1	要旨	1
1.2	論文の構成	1
第 2 章	Continuation based C(CbC)	2
2.1	Continuation based C(CbC)	2
2.1.1	GCC	2
2.1.2	LLVM/Clang	2
2.1.3	Singularity	2
2.2	CbC on GCC	3
2.3	CbC on GCC CrossCompile	3
2.4	CbC on LLVM / Clang	3
2.5	CbC on LLVM/Clang CrossCompile	3
第 3 章	Raspberry Pi 上の GearsOS	4
3.1	GearsOS	4
3.2	xv6	4
3.3	Raspberry Pi	4
3.4	Raspberry Pi 上の xv6	4
第 4 章	UEFI	6
4.1	UEFI	6
4.2	UEFI vs BIOS	6
4.3	Raspberry Pi UEFI	6
4.4	UEFI Hello World	6
第 5 章	Boot Loader	7
5.1	bootloader	7
第 6 章	今後の課題	8
6.1	今後の課題	8

目 次

表 目 次

ソースコード目次

第1章 初めに

1.1 要旨

2017年にIntel社が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は長年に渡り16bitパソコンの時代からの資産を引き継いできたため、16bitモードでしか動作しない。そのためPCの進化に伴い、致命的な問題点が発生する。問題点として、拡張性がないことがあげられる。EthernetやUSBにつながるでディスクなど、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。

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はレガシーOSなため、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。本研究では、ARMで動くシングルコンピュータであるRaspberryPi上にUEFIからGearsOSをブートさせることを目指している。

1.2 論文の構成

第2章 Continuation based C(CbC)

2.1 Continuation based C(CbC)

Continuation based C(CbC)[3] は、当研究室で開発され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。CbC には、GCC[5] 上に実装されたものと LLVM/Clang[6] 上に実装されたものがある。GCC と LLVM/Clang で実装された CbC を ARM でコンパイルするにはクロスコンパイラが必要になる。このクロスコンパイラ環境を Singularity で作成した。

2.1.1 GCC

GCC は GNU Compiler Collection の略で GNU プロジェクトが開発及び配布している、C/C++/Object-C などのプログラミング言語のコンパイラ集のことである。

2.1.2 LLVM/Clang

LLVM とは、モジュラー構成及び再利用可能なコンパイラとツールチェーン技術などを開発するプロジェクトの名称である。Clang は LLVM をバックエンドとして利用する C/C++/Object-C のコンパイラである。

2.1.3 Singularity

Singularity[7] とは、ユーザーが自身の計算環境を完全再現し、保持できる様にした Linux コンテナである。Singularity はマルチユーザーに対応していて、コンテナ内の権限は実行ユーザーの権限を引き継ぐ。そのため、ユーザーに特別な権限の設定が必要ない。また、複雑なアーキテクチャとワークフローをサポートできるよう設計されていて、ほぼ全ての環境に適応できる。

2.2 CbC on GCC

あああ

2.3 CbC on GCC CrossCompile

aaa

2.4 CbC on LLVM / Clang

あああ

2.5 CbC on LLVM/Clang CrossCompile

aaa

第3章 Raspberry Pi上のGearsOS

3.1 GearsOS

GearsOS[2]は当研究室で信頼性と拡張性をテーマに開発されている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をCbCで置き換えを行っている。これにより、OSのこのシステムコールを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ようになる。

3.2 xv6

xv6[4]とは、マサチューセッツ工科大の大学院生向け講義の教材として使うために、UNIX V6というOSをANSI-C(規格化されたC言語)に書き換え、x86に移植したxv6 OSである。x86アーキテクチャで動作する。xv6はプロセス、仮想メモリ、カーネルとユーザの分離、割り込み、ファイルシステムなどの基本的なUnixの構造を持つにも関わらず、シンプルで学習しやすい。

3.3 Raspberry Pi

Raspberry Piは、ARMプロセッサを搭載したシングルコンピュータ。Raspberry Piにはいくつか種類があり、本研究ではRaspberry Pi 3 Model Bを仕様する。Raspberry Pi 3 Model Bには、USB2.0コネクタが4つ、microSDカードスロット、HDMI出力、40ピンGPIOなどがついている。CPUは、ARMアーキテクチャのCortex-A53でCPUクロックは1.4GHzである。さらにメモリは1GBある。

3.4 Raspberry Pi 上の xv6

xv6はx86で動作するOSである。Raspberry Pi上でxv6を動かすためには、ARMに対応したxv6を用意する必要がある。そのためRaspberry Pi用に移植したxv6-rpiを用いる。Raspberry Pi上で起動しているxv6に入力を行うためにUSBシリアルケーブルでMacBookと接続する。その時、Raspberry Piの6番ピン(黒)、8番ピン(白)、10番ピン

(緑)の3つをUSBシリアルケーブルで接続する。この時、HDMIでディスプレイに接続しておく。Mac側では、USBシリアルケーブルのドライバをインストールしてRaspberry Piと接続すると、devディレクトリ直下にtty.usbserialとして認識される。Mac側でscreenコマンドを使い、シリアル通信を行うと、Macのキーボードから入力を行えるようになる。この時、screenコマンドを打ってから、Raspberry Piに電源を入れないと正常に起動しない。

第4章 UEFI

4.1 UEFI

UEFIとは、Unified Extensible Firmware Interfaceの略で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。1990年代半ばにIntelによってBIOSに変わるEFI仕様が開発された。2005年にIntel、AMD、Apple、Microsoftなどの企業からなるUnified EFI Forumという業界団体のもとUEFIが開発された。UEFIは単なるインタフェースの仕様であるため、特定のプロセッサに依存しない。以前までのBIOSと異なり、近代的なソフトウェア開発手法を用いることが推奨されていて、C言語などで実装ができる。

4.2 UEFI vs BIOS

aaa

4.3 Raspberry Pi UEFI

Raspberry PiはARMプロセッサを搭載しているので、ARMで動くUEFIを用意する必要がある。UEFIはマルチプラットフォームであり、UEFIの仕様はハードウェアに依存しないため、異なるCPUアーキテクチャでも動作できる。

4.4 UEFI Hello World

UEFIを開発する際に、

第5章 Boot Loader

5.1 bootloader

aaa

第6章 今後の課題

6.1 今後の課題

aaaa

参考文献

- [1] Intel/Unified EFI Forum, <https://www.uefi.org> ,2017/11/3.
- [2] 清水隆博, 河野真治. GearsOS のメタ計算. 琉球大学工学部情報工学科令和 3 年度学位論文 (修士),2021.
- [3] 宮城光希, 河野慎治. CbC 言語による OS 記述. 琉球大学工学部情報工学科平成 29 年度学位論文 (学士),2017.
- [4] Russ Cox, M Frans Kaashoek, and Robert Morris. Xv6, a simple unix-like teaching operating system,2012.
- [5] GNU Compiler Collection (GCC) Internals,<https://gcc.gnu.org/onlinedocs/gccint/>
- [6] Clang: a C language family frontend for LLVM,<https://clang.llvm.org>.
- [7] <https://sylabs.io/singularity/>

謝辞

感謝します。

2021年2月
奥田 光希