

継続を基本とした OS Gears OS

清水 隆博^{1,a)} 河野 真治^{2,b)}

概要：継続を基本とする C と互換性のある言語、Continuation Based C (CbC) を用いて OS の実装を考案した。状態遷移単位で OS の処理を実装することで、処理の入出力が明確化され、定理証明支援系に適した表現形式で処理が記述可能である。現在 CbC を用いて開発している OS、GearsOS は Xv6 をベースに実機での動作を目指している。ここでは現在の GearsOS の状況、今後の展望について考察する。

キーワード：システムプログラミング, CbC, 軽量継続, OS, CMake

1. 証明可能な OS

コンピュータ上で動作するあらゆるソフトウェアや資源を管理する OS は、高い信頼性が保証されてほしい。信頼性の保証にはテストプログラムを用いた検証や、形式手法を用いた証明を使う手法が存在する。頻繁に並列処理を行う OS では、スレッド間の共通資源の競合などの非決定的な実行を行う。このため、OS の信頼性を保証する上で、テストやデバッグを用いる手法では、発生している状態を完全に保証するのは困難である。

テストを用いる方法ではなく、形式手法的なアプローチを用いて OS の信頼性を保証したい。そのためには定理証明支援系などで証明が可能な形式と、等価な形式で OS を記述する必要がある。現在開発している GearsOS は、継続を基本とする言語 Continuation Based C (CbC) で実装されている。CbC は状態遷移単位での実行であり、他の状態に遷移する際に今までの環境を持たない。

CbC で実装した処理は入出力が明確化され、定理証明支援系で表現可能な形式にする事が可能である。

2. Continuation Based C

Continuation Based C (CbC) とは GearsOS の記述に利用しているプログラミング言語である。C 言語の下位言語として設計されており、C コンパイラである GCC、LLVM/Clang 上に実装が存在する。CbC は通常の関数呼び出しとは異なり、軽量継続を基本としている。通常 C の関数呼び出しでは、call 命令により、スタックポインタを操作し、ローカル変数や、レジスタ情報をスタックに保存する。CbC の軽量継続は、アセンブラレベルでは jmp で表現され、スタックフレームを操作することなく次の状態に遷移する。CbC の状態は CodeGear と呼ばれる単位で記述される。

3. GersOS の基本単位

実行単位としては CbC で導入された CodeGear を用いる。CodeGear は関数よりも単位が小さく、かつアセンブラよりも単位が大きく処理を記述す

¹ 琉球大学大学院理工学研究科情報工学専攻

² 琉球大学工学部工学科知能情報コース

a) anatofuz@cr.ie.u-ryukyu.ac.jp

b) kono@ie.u-ryukyu.ac.jp

ることが可能である。そのため、OSの必要な資源管理などのメタ計算を記述するのに適していると考えられる。

GearsOSでは使われる情報を、DataGearと呼ばれる単位で構成する。DataGearはCの構造体のように宣言するが、すべてのDataGearはContextと呼ばれるデータ構造の中で、共用体として管理されている。CodeGearでは入出力をDataGearで管理している。CodeGearの入力で使用されるDataGearを、InputDataGearと呼び、出力するDataGearをOutputDataGearと呼ぶ。この入出力の組をTaskとして定義し、InputDataGearの依存関係が解決されたTaskから、CodeGearが並列実行される。

4. GearsOSで記述されたxv6

GearsOSの機能であるContextなどを用いて、実際に実機で動作するOSを作成したい。実機で動作するOSのベース実装として、システムコールなどのシンプルなUNIXの機能を持つxv6に着目した。xv6はARMプロセッサを持つRaspberryPi上で動作する、xv6_rpiというバリエーションが存在する。GearsOSを実行で動作させるために、xv6_rpiのソースコードをGearsOSで一部再実装している。現在はxv6のプロセスであるproc構造体に、GearsOSのcontextを導入し、GearsOSとしてもxv6としても解釈可能な形で開発している。

5. GearsOSのクロスコンパイル

GearsOSはRaspberryPi上での動作を目指している。RaspberryPiはARMのCPUが搭載されている為、動作にはARMのバイナリファイルが必要となる。しかしRaspberryPiを利用してGearsOS自身のビルドを行うと、マシンパワーの問題でビルドに莫大な時間が掛かってしまう。著者らが使うことが多い、資源が潤沢なx86マシンから、ARMにクロスコンパイルする必要がある。GCC上に実装しているCbCコンパイラは、ARMを出力するようにコンパイラを再構築する必要があった。他方LLVM/clang上に実装しているCbCコンパイラは、ARMのライブラリは必

要であるものの、本体を再度ビルドすることなくクロスコンパイラとして利用可能である。今回はRaspberryPiのデフォルトOSであるRaspbianから、ARMのライブラリをx86マシン上に転送し、LLVM/clang上に実装したCbCコンパイラを用いてビルドした。ビルドツールとしてはCMakeを導入している。CMakeでクロスコンパイルを行う際に、クロスコンパイラなどを引数で指定する必要がある為、引数の解決に一部Perlスクリプトを利用している。

6. 今後の課題

現状はxv6をGearsOSとして書き直している段階であり、システムコールで呼び出された後のkernel部分の処理を順次Interfaceとして実装している。RaspberryPi上で動作する様にクロスコンパイルをする環境はCMakeを利用して構築出来たので、実際にRaspberryPi上でInterfaceを導入したGearsOSを動作させる必要がある。またxv6はUEFIでのブートが組み込まれているので、これを実装したい。UEFIでブートが可能になると、各種デバイスドライバを組み込むのが容易になる為、USB3.0の規格であるxHCIなどをxv6上に実装することが可能となる。xHCIを実装する事によってxv6を実機で動かした際に、USB接続をしたキーボードが使用可能となる。これらの実装には、CbCで実装された実装としても使用可能な仕様記述言語を用いる予定である。また、実際にxv6上での処理を定理証明支援系などで証明を行い、証明しやすい実装と、処理に適した実装にInterfaceを通して切り替える機構を実装することも課題である。

参考文献

- [1] 宮城光希, 桃原 優, 河野真治: Gears OS のモジュール化と並列 API, 技術報告 11, 琉球大学大学院理工学研究科情報工学専攻, 琉球大学大学院理工学研究科情報工学専攻, 琉球大学工学部情報工学科 (2018).
- [2] 並列信頼研究室: CbC_gcc, 琉球大学 (online), available from (http://www.cr.ie.u-ryukyu.ac.jp/hg/CbC/CbC_gcc/) (accessed 2018-11-21).