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、 Conitinuation 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、 発生しう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 を記述する必要性がある。現在開発している GearsOS は、 継続を基本とする言語 Conitinuation 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。

[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

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

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).