

# CbC 言語による OS 記述

135756F 氏名 宮城光希 指導教員：河野 真治

## Abstract

## 1 メタ計算の重要性

プログラムを記述する際、通常の処理の他に、メモリ管理、スレッドの待ち合わせやネットワークの管理、エラーハンドリング等、記述しなければならない処理が存在する。これらの計算を Meta Computation と呼ぶ。

Meta Computation を通常の計算から切り離して記述するためには処理を細かく分割する必要がある。しかし、関数やクラスなどの単位は容易に分割できない。

そこで当研究室では Meta Computation を柔軟に記述するためのプログラミング言語の単位として Code Gear、Data Gear という単位を提案している。

Code Gear は関数に比べて細かく分割されているので Meta Computation をより柔軟に記述できる。Code Gear、Data Gear にはそれぞれメタレベルの単位である Meta Code Gear、Meta Data Gear が存在し、これらを用いて Meta Computation を実現する。

Continuation based C (CbC) はこの Code Gear 単位を用いたプログラミング言語として開発している。

CbC は軽量継続による遷移を行うので、継続前の Code Gear に戻ることはなく、状態遷移ベースのプログラミングに適している。

また、当研究室で開発している Gears OS は Code Gear、Data Gear の単位を用いて開発されており、CbC で記述されている。

本研究では CbC を用いての Gears OS の実装と CbC におけるユーザーの関知しない Meta Computation の自動生成を行なう。

## 2 Continuation based C (CbC)

CbC は処理を Code Gear とした単位を用いて記述するプログラミング言語である。Code Gear は入力と出力を持ち、CbC では引数が入出力となっている。Code Gear から次の Code Gear へと goto による継続で遷移で処理を行い、引数として出力を与える。図 1 は Code Gear 間の処理の流れを表している。

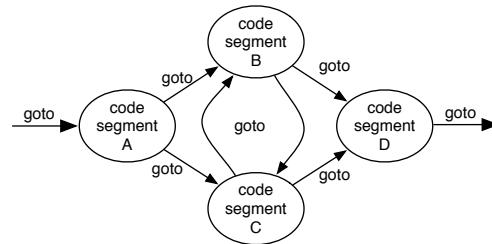


図 1: goto による code gear 間の継続

## 3 Gears OS

Gears OS では並列実行するための Task を、実行する Code Gear、実行に必要な Input Data Gear、Output Data Gear の組で表現する。Data Gear はデータの単位であり、int や文字列などの Primitive Type を持っている。Code Gear は任意の数の Input Data Gear を参照して処理を行い、Output Data Gear を出力し処理を終える。また、接続された Data Gear 以外には参照を行わない。処理やデータの構造が Code Gear、Data Gear に閉じているため、これにより実行時間、メモリ使用量などを予測可能なものにするのが可能になる。

Gears OS では Meta Computation を Meta Code Gear、Meta Data Gear で表現する。Meta Code Gear は通常の Code Gear の直後に遷移され、Meta Computation を実行する。

CbC は Code Gear を処理の単位として用いたプログラミング言語であるため、Gears OS の Code Gear を記述するのに適している。

図 2 に Gears OS の構成図を示す。

## 参考文献

- [1] TightVNC Software. <http://www.tightvnc.com>.
- [2] RICHARDSON, T., STAFFORD-FRASER, Q., WOOD, K. R., AND HOPPER, A. virtual network computing, jan 1998.
- [3] Yu TANINARI and Nobuyasu OSHIRO and Shinji KONO. Vnc を用いた授業用画面共有システムの実装と

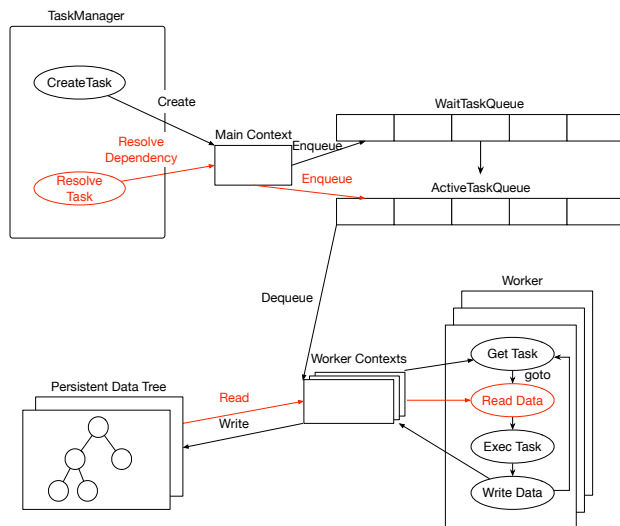


図 2: Gears OS の構成図

設計. 日本ソフトウェア科学会第 28 回大会論文集, sep 2011.

- [4] Tatsuki IHA and Shinji KONO. 有線 lan 上の pc 画面配信システム treevnc の改良. 第 57 回 プログラミング・シンポジウム, jan 2016.